Universidade Federal do Ceará
Centro de Tecnologia
Departamento de Engenharia Elétrica
Trabalho de Conclusão de Curso

# Interface MATLAB – PSCAD para Simulação Integrada

Guilherme Martins Gomes Nascimento

Fortaleza, junho de 2011

**Monografia | Guilherme Martins Gomes Nascimento | V.F. || 2011**

GUILHERME MARTINS GOMES NASCIMENTO

# Interface MATLAB – PSCAD para Simulação Integrada

Monografia submetida à Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de Engenheiro Eletricista.

Orientador: Profª. Ph.D.Ruth Pastôra Saraiva Leão

Fortaleza
Junho de 2011

INTERFACE MATLAB – PSCAD PARA SIMULAÇÃO INTEGRADA

Esta monografia foi julgada adequada para obtenção do título de Engenheiro Eletricista e aprovada em sua forma final pelo programa de Graduação em Engenharia Elétrica na Universidade Federal do Ceará.

Guilherme Martins Gomes Nascimento

Banca Examinadora:

Profª. Ph.D. Ruth Pastôra Saraiva Leão,
Orientadora

Prof. MSc. Nelber Ximenes Melo

Engª. Janaína Barbosa Almada,

Fortaleza, Junho de 2011

*“Não to mandei eu? Esforça-te, e tem bom ânimo; não temas, nem te espantes; porque o SENHOR teu Deus é contigo, por onde quer que andares.”*

*(Josué 1.9)*

*“...no mundo tereis aflições, mas tende bom ânimo, Eu venci o mundo”*

*(João 16:33)*

*A Deus,*

*A meus pais, Robério e Tirza,*

*A meus irmãos, Rafael e Gabriel,*

*A minha amada, Aline*

*A meus amigos*

Nascimento, G. M. N. “Interface MATLAB – PSCAD para Simulação Integrada”, Universidade Federal do Ceará – UFC, 2011, 90p

Esta monografia tem por objetivo apresentar a integração de dois softwares de simulação amplamente utilizados em projetos de engenharia, o PSCAD e o MATLAB. Ambos os softwares possuem peculiaridades que são de grande utilidade para alguns destes projetos, se diferenciando pelas funções apresentadas ao usuário. Se essas diferentes funções puderem ser utilizadas de maneira conjunta, as simulações desenvolvidas podem trazer grandes vantagens. Muitos obstáculos foram encontrados durante a instalação e integração no software. Por isso, foi descrito de forma objetiva esses problemas, expondo as soluções encontradas para eles, de modo a auxiliar àqueles que desejem utilizar-se dessa interface. As configurações de simulação são explicadas e exemplificadas, tanto no PSCAD como no MATLAB, uma vez que essas configurações devem estar condizentes, já que os softwares devem ser capazes de reconhecer os dados recebidos e tratá-los devidamente para enviar dados coerentes com a programação do outro software. Ainda são apresentadas algumas ferramentas do PSCAD, tanto a fim de aprimorar os conhecimentos sobre este software de grande contribuição nas análises de sistemas de potência, como também solucionar problemas de tempo de simulação, que se mostrou demasiadamente grande em alguns casos. Se o tempo de simulação não for de grande importância em um projeto, a interface pode chegar a substituir estas ferramentas.

Palavras-Chave: Integração, PSCAD, MATLAB, Simulação integrada.

Nascimento, G. M. N. “Matlab – PSCAD software interface for integrated simulation”, Universidade Federal do Ceará – UFC, 2011, 90p

This work aims to present the integration of two simulation softwares widely used in engineering projects: PSCAD and MATLAB. Both softwares have peculiarities that are very useful to some of the projects, being different from each other on the functions presented to the users. If those functions can be used in a joint manner, the developed simulation capacities can be greatly enhanced. Many obstacles were found during the softwares installation and integration processes, so these problems were described objectively, exposing the solutions in order to assist those willing to use this interface. PSCAD and MATLAB simulation configurations are explained and exemplified, once the configuration need to be flawless, because these softwares need to recognize the data sent and received correctly. Some PSCAD tools are also presented in order to improve the understanding of this software so it is easier to reduce problems with the integrated simulation time, which, in some cases, proved to be too great. If the simulation time isn’t very important for a project, the interface can replace these tools.

Keywords: Integration, PSCAD, MATLAB, Joint simulation.

# Sumário

**LISTA DE FIGURAS** ..... **xi**

**LISTA DE TABELAS** ..... **xii**

**CAPÍTULO 1**
**INTRODUÇÃO** ..... **1**

**CAPÍTULO 2**
**HISTÓRICO E JUSTIFICATIVAS DE SOFTWARES UTILIZADOS E AQUISIÇÕES** ..... **5**

2.1. CONSIDERAÇÕES INICIAIS ..... 5
2.2. JUSTIFICATIVA DA ESCOLHA DOS SOFTWARES ..... 6
2.3. HISTÓRICO DO PROCESSO DE AQUISIÇÃO E INSTALAÇÃO DE SOFTWARES PARA A INTERFACE ..... 7
2.4. CONSIDERAÇÕES FINAIS ..... 14

**CAPÍTULO 3**
**DESCRIÇÃO E INSTALAÇÃO DOS COMPONENTES NECESSÁRIOS PARA A COMPLETA INTEGRAÇÃO ENTRE OS SOFTWARES** ..... **16**

3.1. CONSIDERAÇÕES INICIAIS ..... 16
3.2. PROCESSO DE INSTALAÇÃO DE TODOS OS COMPONENTES ..... 16
3.3. CONFIGURAÇÕES E SOLUÇÃO DE PROBLEMAS ..... 19
*3.3.1. Verificação das variáveis de ambiente* ..... *21*
*3.3.2. Verificar reconhecimento do PSCAD* ..... *22*
*3.3.3. Verificação da instalação do compilador* ..... *22*
*3.3.4. Verificação do arquivo Fortran_compilers.props* ..... *23*
*3.3.5. Verificação da versão do Microsoft Visual Studio* ..... *24*
*3.3.6. Contato com Suporte Técnico* ..... *25*
*3.3.7. Configurações de integração.* ..... *25*
3.4. CONSIDERAÇÕES FINAIS ..... 27

**CAPÍTULO 4**
**FERRAMENTAS E MÉTODOS DE INTEGRAÇÃO ENTRE OS SOFTWARES** ..... **29**

4.1. CONSIDERAÇÕES INICIAIS ..... 29
4.2. UTILIZAÇÃO DE ARQUIVOS DO MATLAB ..... 30
4.3. PSCAD ..... 30
*4.3.1. Blocos de Componentes* ..... *31*
*4.3.2. Parâmetros* ..... *33*
*4.3.3. Script* ..... *36*
4.3.3.1 Computations ..... 36
4.3.3.2 Branch ..... 37
4.3.3.3 Fortran ..... 38
4.4. INTEGRAÇÃO PSCAD – M-FILE ..... 38
*4.4.1. PSCAD* ..... *39*
*4.4.2. M-FILE* ..... *43*
*4.4.3. Exemplo 1 – Resultados e comentários* ..... *44*
4.5. INTEGRAÇÃO PSCAD – SIMULINK ..... 46
*4.5.1. PSCAD* ..... *46*
*4.5.2. SIMULINK* ..... *47*

*4.5.3. Exemplo 2 – Resultados e comentários ..... 48*
4.6. EXEMPLO PSCAD SUPPORT, DESENVOLVIDO PELO SUPORTE TÉCNICO ..... 50
4.7. CONSIDERAÇÕES FINAIS ..... 54

**CAPÍTULO 5**
**CONCLUSÕES ..... 55**

**REFERÊNCIAS BIBLIOGRÁFICAS ..... 57**

**APÊNDICE A AUXÍLIO DO SUPORTE TÉCNICO DO SOFTWARE PSCAD VIA E-MAILS ..... 58**

**APÊNDICE B CÓDIGO DESENVOLVIDO EM FORTRAN PARA PSCAD PARA EXEMPLO 1 ..... 78**

**APÊNDICE C CÓDIGO DESENVOLVIDO NO M-FILE PARA EXEMPLO 1 ..... 80**

**APÊNDICE D CÓDIGO DESENVOLVIDO EM FORTRAN PARA PSCAD PARA EXEMPLO 2 ..... 82**

**ANEXO A INTEGRAÇÃO MATLAB – PSCAD ..... 84**

**ANEXO B CÓDIGO EM FORTRAN PARA PSCAD PARA EXEMPLO DO SUPORTE TÉCNICO ..... 89**

# Lista de Figuras

Figura 1.1 - Exemplo de comunicação entre softwares ..... 3
Figura 2.1 - Aplicativo FortranMedic ..... 9
Figura 2.2 - VMWare Player - Emulação de Máquinas Virtuais ..... 13
Figura 3.1 - Fluxograma – Instalação ..... 19
Figura 3.2 - Localização Workspace Settings ..... 20
Figura 3.3 – Workspace Settings - Compilador em uso ..... 21
Figura 3.4 - FortranMedic - Versões do PSCAD Instaladas ..... 22
Figura 3.5 - FortranMedic - Verificação do compilador ..... 23
Figura 3.6 - FortranMedic - Verificação do fortran_compiler.props ..... 24
Figura 3.7 - FortranMedic - Versões do Visual Studio ..... 24
Figura 3.8 - Workspace Settings - MATLAB ..... 26
Figura 3.9 - Erro gerado pela falta do M-File ..... 27
Figura 3.10 - Fluxograma - Solução de erros ..... 28
Figura 4.1 - Funcionamento da Interface ..... 29
Figura 4.2 - Link projeto - bibliotecas MATLAB ..... 31
Figura 4.3 - New Component - Criar um novo componente ..... **Erro! Indicador não definido.**
Figura 4.4 - Criação de novo componente ..... 32
Figura 4.5 - Definição da função de pinos ..... 32
Figura 4.6 - Retornar ao modo de construção de circuito ..... **Erro! Indicador não definido.**
Figura 4.7 - Configurações de porta de conexão ..... 34
Figura 4.8 - Parameters - Configuração de campo de entrada ..... 35
Figura 4.9 - Parameters - Configuração de caixa de escolha ..... 35
Figura 4.10 - Segment Manager ..... 36
Figura 4.11 - Exemplo de utilização de Branch ..... 38
Figura 4.12 - Exemplo 01 – Circuito para integração ..... 39
Figura 4.13 - Exemplo M-File - Componente de integração ao MATLAB no PSCAD ..... 39
Figura 4.14 - Exemplo M-File - Parâmetros do componente de integração ..... 40
Figura 4.15 - Exemplo M-File - Sistema completo ..... 41
Figura 4.16 - Exemplo 1 - PSCAD - Simulação completa ..... 45
Figura 4.17 - Exemplo 1 - MATLAB – Gráficos obtidos na simulação completa ..... 45
Figura 4.18 - SIMULINK - Bloco de leitura de dados ..... 47
Figura 4.19 - Exemplo 2 - Circuito SIMULINK ..... 47
Figura 4.20 - SIMULINK - Configurações de Parâmetros - Tempo de simulação ..... 48
Figura 4.21 - Exemplo 2 - Resultados ..... 49
Figura 4.22 - Exemplo de aumento de velocidade de simulação ..... **Erro! Indicador não definido.**
Figura 4.23 - Exemplo Support - Circuito ..... 50
Figura 4.24 - Exemplo Support - Controles e Componente de interface com o SIMULINK ..... 51
Figura 4.25 - Exemplo Support - Parâmetros do componente de integração ..... 51
Figura 4.26 - Exemplo Support - Sistema no SIMULINK ..... 53
Figura 4.27 - Exemplo Support - Sistema simulado - PSCAD ..... 53

# Lista de Tabelas

Tabela 2.1 - Compatibilidade Windows7/PSCAD V4 - MATLAB ..... 11

# Capítulo 1

# Introdução

Na engenharia, modelos de projetos são comumente desenvolvidos a fim de criar um novo produto, serviço ou processo (DUARTE e AVELAR, 2010). No desenvolvimento destes projetos, é comum haver uma grande quantidade de dados a ser analisados, e nem sempre é suficiente prever o desenvolvimento do projeto como um todo somente a partir de conhecimentos teóricos. Muitas vezes, problemas ou erros no desenvolvimento do projeto passam despercebidos, seja por desatenção do projetista, ou pela complexidade do projeto, que conta com enorme quantidade componentes e dados, necessitando repetidas análises para diferentes cenários e condições de operação, o que torna humanamente inviável o completo acompanhamento e análise prévia do projeto. Verifica-se assim uma necessidade de alguma ferramenta para observar o projeto, analisá-lo e certificar sua viabilidade antes de colocá-lo em prática.

Existem softwares que se utilizam de esforço computacional para fazer todas estas análises, que fazem uso de algoritmos e conceitos teóricos previamente programados, simulando o desenvolvimento do projeto, de acordo com o que for definido, tanto pelo software como pelo usuário. A simulação funciona como uma observação do comportamento do sistema em teste em condições definidas. Obviamente, simular projetos sem todo um estudo prévio se mostra extremamente ineficaz, uma vez que não há um embasamento teórico necessário para definir o problema.

Os softwares de simulação são utilizados a fim de verificar se o que foi concebido se comporta como o esperado, não desperdiçando tempo ou esforço ao implementar fisicamente algo que poderá não funcionar corretamente. Eles disponibilizam uma maneira simplificada de observar o comportamento do projeto, em suas condições definidas, ajudando a realizar análises de erros e até modificações no projeto a fim de aprimorar e eficientizar os processos, podendo também eliminar gastos possivelmente desnecessários.

Dessa forma, a simulação se mostra algo de extrema importância no desenvolvimento de projetos.

Alguns destes softwares de simulação, entretanto, nem sempre dispõem de todas as funcionalidades desejadas pelo usuário. Nestes casos, é possível que exista outra ferramenta que possa substituir a primeira, suprindo suas faltas.Quando isso ocorrer, supondo que a primeira ferramenta de simulação já foi utilizada por algum tempo, várias simulações já concluídas utilizando o primeiro software se tornarão obsoletas, uma vez que o segundo passará a ser utilizado. Como não é comum duas ferramentas de simulação suportarem o mesmo tipo de arquivo, ou haver métodos de comunicação entre softwares distintos, nada daquilo que já foi feito no primeiro software poderá ser aproveitado no segundo.

Ainda existe a outra possibilidade de dois softwares distintos se complementarem, de modo que um tenha certas funções que o outro não possui. Muitas vezes o usuário tem mais conhecimentos em um software, o que facilita seu trabalho ao utilizar tal software, trabalhando de maneira mais completa. Entretanto, ao verificar que este software não possui certa funcionalidade, percebe que existe outro software com o qual ele está menos familiarizado que a possui. Isso mostra como a integração entre dois softwares pode ser interessante.

A idéia inicial dessa integração seria criar um canal de comunicação entre ambos os softwares, fazendo com que cada um utilize seu ponto forte a favor da simulação, criando assim uma simulação mais completa.

Este canal de comunicação serviria para que os softwares se comunicassem, enviando e recebendo informações, dados de comunicação. Ao compartilhar essas informações, cada lado da simulação faria sua parte e o uso compartilhado de recursos e de aproveitamento de código traria benefícios ao usuário.

Basicamente, quando um lado da simulação receber dados do outro, deverá interpretar esses dados e utilizá-los como dados de entrada na sua simulação, enviando assim para o outro os dados de saída, como uma resposta. O mesmo ocorreria com o segundo lado da simulação, sendo criando assim um ciclo que se complementaria, e seguiria assim até o término da simulação como um todo.

**Figura 1.1 - Exemplo de comunicação entre softwares**

Essas informações devem ser completamente compatíveis com ambos os programas de simulação, possibilitando assim uma comunicação completa, sem problemas de interpretação. Caso haja algum problema na interpretação dos dados, um dado de um software pode ser interpretado incorretamente pelo outro software, podendo assim gerar respostas completamente incoerentes com o que se deseja simular, acabando assim com todo o propósito da simulação.

Como não é comum haver essa comunicação entre softwares, é possível que existam outros meios de obtê-la, como, por exemplo, softwares desenvolvidos especificamente para essa função.

Outro problema que se deve atentar é o fato de que uma simulação apenas pode exigir grande esforço computacional. Ao utilizar essa integração, ambos os softwares estariam simulando simultaneamente, exigindo um esforço computacional maior ainda. Nem todo computador possui tal capacidade computacional, o que poderia limitar a simulação, levando a simulações de duração extremamente elevada, ou até mesmo travamentos do sistema operacional, sendo necessário a reinicialização da máquina.

Dessa maneira, este trabalho procura apresentar uma alternativa para essa idéia. O Capítulo 2 expõe a motivação deste trabalho, a escolha dos softwares a serem integrados, expondo justificativas para essa escolha, além de todo o processo ocorrido até a obtenção da interface funcional entre os softwares.

Então, o Capítulo 3 entra em detalhes sobre os processos de instalação, expondo problemas e softwares necessários para que esta instalação seja dada com sucesso. Também relata uma lista de problemas obtidos para que os softwares se reconhecessem, com a intenção de auxiliar àqueles que possam passar por problemas semelhantes.

Após isso, o Capítulo 4 expõe algumas ferramentas que poderão auxiliar essa integração. Também mostra quais processos devem ser seguidos para obter a comunicação entre os softwares, utilizando exemplos para auxiliar a explicação do modo de funcionamento da interface. Mostra ainda possíveis alternativas para se obter resultados em um tempo de simulação melhor.

Finalmente, conclui-se o trabalho no Capítulo 5, apresentando as conclusões obtidas a partir do trabalho, alguns dos principais pontos discutidos, seguidos por algumas sugestões para desenvolvimento de trabalhos futuros nesta linha de pesquisa.

# Capítulo 2

# Histórico e Justificativas de Softwares Utilizados e Aquisições

## 2.1. Considerações Iniciais

Na engenharia é comum o uso de softwares de simulação uma vez que geralmente não é viável realizar estudos, análises e testes no sistema real.

Na Engenharia Elétrica há vários softwares que variam segundo a área de aplicação, a interface com o usuário, as facilidades oferecidas pelo software, o número de componentes disponível na biblioteca e o tamanho do sistema que são capazes de representar.Na área de Sistemas de Potência, por exemplo, pode-se citar a utilização na atuação da proteção de aerogeradores, ou na operação de microrredes e seus modos de funcionamento.

Com o crescimento do uso de fontes renováveis para geração de energia elétrica, torna-se importante lançar mão de ferramentas de simulação que disponham de modelos para simulação computacional de sistemas elétricos.

Os softwares PSCAD/EMTDC® e Matlab®/Simulink são duas ferramentas amplamente usadas na engenharia elétrica e em particular no curso de engenharia elétrica da Universidade Federal do Ceará (UFC).O PSCAD/EMTDC® é um software, largamente utilizado em todo o mundo, para simulação de transitório eletromagnético em sistemas de potência, desenvolvido pela empresa *Manitoba HVDC Research Center*, em Manitoba, Canadá. O software SIMULINK/MATLAB® oferece uma grande capacidade de criar sistemas de controle complexos, a possibilidade de simular sistemas elétricos, entre muitas outras funcionalidades que não se encaixam no escopo deste trabalho. O SIMULINK ainda apresenta a capacidade de utilizar o sistema de simulações matemáticas oferecido pelo MATLAB, o que incrementa muito sua capacidade de simulação, uma vez que funções matemáticas podem ser utilizadas. (USKI SANNA, 2004) (FARHAD SHAHNIA)

Estas ferramentas são amplamente utilizadas não somente devido às suas capacidades de simulação, mas também pelo bom retorno que ambas têm dado, além da experiência em sua utilização.

## 2.2. Justificativa da escolha dos softwares

O software SIMULINK, do MATLAB, oferece um enorme número de recursos para serem utilizados. Estes recursos são configuráveis, além de abertos a modificações. A capacidade de utilizar os recursos matemáticos presentes no MATLAB leva a uma maior simplicidade de manipulação de dados, seja para análise ou para modificação. As caixas de ferramentas presentes no SIMULINK mostram-se simples de ser utilizadas, necessitando apenas a configuração de alguns parâmetros essenciais à simulação.

O software PSCAD mostra grande capacidade de simulação de sistemas de potência. Este possui uma interface gráfica que permite a adição de blocos como componentes do sistema elétrico (USKI SANNA, 2004). Na realidade, cada um destes blocos é um código, na linguagem de programação Fortran, que se comporta de acordo com aquilo que foi programado. A modificação e até mesmo a visualização desse código é bloqueada pelos desenvolvedores, o que pode limitar sua utilização em alguns casos. O número de blocos também não é tão grande quanto a quantidade presente no SIMULINK. O PSCAD não fornece em sua biblioteca principal todos os tipos de fontes de geração, no entanto, os blocos presentes no PSCAD mostram-se altamente configuráveis, uma vez que existem vários parâmetros a serem modificados. O grau de complexidade de configuração de alguns destes blocos, entretanto, pode ser demasiado para algumas aplicações, uma vez que estas podem ser simuladas em outros softwares de simulação com uma menor complexidade.

Dessa maneira, pode-se observar que a integração desses dois softwares pode ser de grande utilidade. No MATLAB/SIMULINK, a simplicidade de manipular dados usando o MATLAB, a maior simplicidade de configuração de blocos, maior quantidade de blocos e a liberdade de modificações se destacam, enquanto no PSCAD, a capacidade de simulação de sistemas de potência complexos, além da possibilidade de modificação de parâmetros em tempo real de simulação se destacam. Ao integrar ambos os softwares de maneira a criar uma comunicação entre si, poderiam ser utilizados os pontos fortes de cada um destes.

Essa comunicação não pode ser feita de maneira simplificada, uma vez que não existem meios de comunicação pré-programados entre estes softwares. Com esta idéia em mente, pesquisou-se então como essa comunicação entre softwares poderia ser feita.

O manual do usuário do PSCAD/EMTDC, (MANITOBA, 2005), deixa claro que essa integração é possível, porém deve-se utilizar um compilador diferente. Ao instalar o PSCAD, o compilador gratuito *EGCS/GNU Fortran 77 Compiler* é instalado. Esse compilador oferece apenas as funções básicas de simulação do software, sendo ele o responsável pela compilação do código em Fortran. O manual do usuário também sugere que, caso deseje-se fazer simulações mais complexas, deve-se utilizar um compilador mais eficiente, sugerindo o compilador Intel Visual Fortran 9.0.x, vendido pela empresa.

O manual também menciona outro compilador, o Compaq Visual Fortran 6.x, que pode ser utilizado para casos mais complexos que o compilador gratuito suporta.

Vale mencionar que apenas um compilador é utilizado pelo PSCAD, substituindo assim os outros. Logo, os compiladores mais completos são capazes de executar todas as funções dos anteriores, com vantagens em relação a estes.

## 2.3. Histórico do processo de aquisição e instalação de softwares para a interface

Visando a integração entre os softwares PSCAD e Matlab, o compilador Compaq Visual Fortran foi obtido e instalado, a fim de testar sua capacidade de integração com o MATLAB. Infelizmente, não se obteve sucesso com a utilização deste compilador, uma vez que foram encontrados vários problemas em sua instalação e utilização.

Entrou-se em contato, então, com o departamento de vendas do PSCAD, verificando se esta integração era realmente possível. A resposta obtida foi que o compilador Intel Visual Fortran seria o mais recomendado para a completa integração entre os softwares desejados. Com isso, o compilador foi adquirido.

Decorrido algum tempo após a aquisição do software, não se havia obtido sucesso com a integração. Dada a importância de operacionalizar a comunicação do software PSCAD com o

Matlab para os pesquisadores do Grupo de Processamento de Energia e Controle (GPEC) do Departamento de Engenharia Elétrica (DEE) da UFC, um projeto de monografia foi definido com o objetivo de viabilizar a comunicação entre os softwares uma vez que foi afirmado pelas empresas que venderam os softwares PSCAD e Compilador Intel Fortran que a integração era possível.

Iniciou-se então um processo de tentativa e erro para tentar instalar com sucesso o compilador.

Inicialmente, o PSCAD não conseguia "enxergar" que o compilador estava instalado. Após muito tempo tentando fazê-lo enxergar o programa em mais de uma máquina, utilizando mais de um sistema operacional, decidiu-se então entrar em contato com o suporte técnico do software PSCAD, que também é responsável pelo compilador Intel Visual Fortran, através do e-mail de contato: *support@pscad.com*.Todos os e-mails trocados com o suporte técnico encontram-se anexados no apêndice A.

Depois de feitas as devidas introduções aos problemas obtidos até então, finalmente o suporte técnico respondeu o email sugerindo a utilização de um aplicativo por eles desenvolvido, nomeado de FortranMedic (disponível para download em sua versão mais atual em http://pscad.com/updater/utilities/FortranMedic.zip em 27 de junho de 2011), Figura 2.1, que teria a finalidade de gerar um relatório baseado nos registros do PSCAD, de compiladores e próprio sistema operacional, além de possibilitar algumas alterações, preferencialmente com o acompanhamento do suporte técnico. Basicamente, o uso recomendado pelo suporte foi:

- Gerar o relatório;
- Verificar problemas em vermelho;
- Clicar com o botão direito nos problemas que surgiram para o aplicativo corrigi-lo automaticamente.
- Caso o PSCAD ainda não reconheça o compilador, anexar o arquivo do relatório gerado pelo aplicativo em um e-mail e enviá-lo para o suporte técnico.

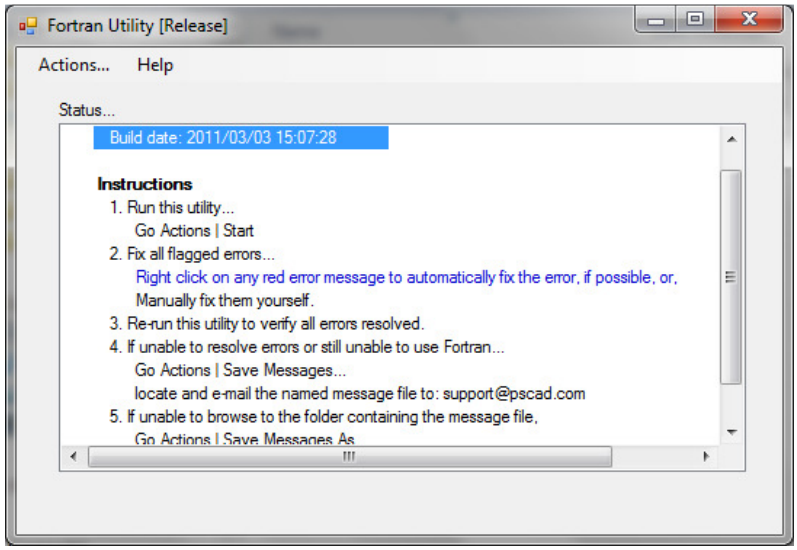

Fortran Utility [Release]
Actions...
Help
Status...
Build date: 2011/03/03 15:07:28
Instructions
1. Run this utility...
Go Actions | Start
2. Fix all flagged errors...
Right click on any red error message to automatically fix the error, if possible, or,
Manually fix them yourself.
3. Re-run this utility to verify all errors resolved.
4. If unable to resolve errors or still unable to use Fortran...
Go Actions | Save Messages...
locate and e-mail the named message file to: support@pscad.com
5. If unable to browse to the folder containing the message file,
Go Actions | Save Messages As

**Figura 2.1 - Aplicativo FortranMedic**

.

Através deste aplicativo, o suporte técnico obteve as informações que necessitava para auxiliar na instalação de todos os softwares relacionados à integração.

Inicialmente, ao resolver os problemas marcados no aplicativo FortranMedic, o software PSCAD passou a reconhecer o compilador Intel Visual Fortran instalado no computador, porém, ao tentar utilizá-lo em uma simulação, foi obtido um erro.

Observando que a maior parte dos problemas no relatório gerado foi relacionada à instalação de um software fornecido pela Microsoft, o Microsoft Visual Studio, que seria um pacote de desenvolvimento em linguagens de programação da Microsoft(VISUALSTUDIO, 2011), pesquisou-se na internet onde obtê-lo, a fim solucionar os problemas a ele relacionados.

Procurou-se então reinstalar o software, incluindo outro software necessário para sua instalação, o .NET Framework 4.0, que seria um pacote que reúne as linguagens de programação da Microsoft, permitindo assim sua utilização no sistema operacional (FRAMEWORK, 2011).

Sem sucesso, foi retomado o contato com o suporte técnico, que respondeu que o antigo compilador Compaq Fortran ainda estava instalado ou havia deixado resquícios de sua instalação no registro do Windows e até mesmo em suas pastas. Para tal problema, foram dadas três opções de solução:

- Deletar a pastas remanescentes, o que poderia causar problemas no funcionamento de outros programas;
- Mover manualmente o nível de prioridade das variáveis de ambiente relacionadas ao compilador;
- Utilizar a nova versão do aplicativo FortranMedic para mover tais variáveis de maneira simplificada.

Utilizando o novo FortranMedic para corrigir o problema, que foi aprimorado para esta aplicação, bastou localizar as varáveis desejadas na lista gerada pelo aplicativo e, pressionando o botão direito do mouse sobre elas, movê-las para o fim da lista de variáveis de ambiente, lhes dando, assim, maior prioridade (visto que o padrão do FortranMedic é de modo que os de maior prioridade se encontram no fim da lista). O processo completo descrito pelo suporte técnico pode ser visto no apêndice A.

As variáveis que deveriam ser realocadas para o fim da lista são:

Na subdivisão PATH Environment Variable

- c:\program files\microsoft visual studio\common\tools
- c:\program files\microsoft visual studio\common\msdev98\bin
- c:\program files\microsoft visual studio\df98\bin
- c:\program files\microsoft visual studio\vc98\bin

Na subdivisão LIB Environment:

- C:\Program Files\Microsoft Visual Studio\DF98\IMSL\LIB
- C:\Program Files\Microsoft Visual Studio\DF98\LIB
- C:\Program Files\Microsoft Visual Studio\VC98\LIB

Após realocadas, deve-se confirmar suas novas posições rodando novamente o FortranMedic. Caso isso não resolvesse o problema, dever-se-ia realocar as seguintes variáveis utilizando o aplicativo novamente. Na subdivisão PATH Environment Variable:

- C:\Program Files (x86)\Intel\Compiler\11.1\067\lib\Intel64
- C:\Program Files (x86)\Intel\Compiler\11.1\067\mkl\em64t\bin
- C:\Program Files (x86)\Intel\Compiler\11.1\067\lib\Intel64

Com isso, foi corrigido o problema de instalação do compilador, uma vez que ele permitiu a simulação de exemplos simples.

Ao instalar o software PSCAD, alguns arquivos exemplos são disponibilizados para exemplificar o uso das diversas aplicações do software. Conhecendo isto, buscou-se um exemplo que se utilizava da integração MATLAB-PSCAD a fim de verificar o completo funcionamento do compilador. Entretanto, mais uma vez observaram-se erros ao tentar simular um destes exemplos.

Após mais alguma pesquisa sem soluções aparentes, entrou-se em contato novamente com o suporte técnico. A resposta obtida foi que uma vez que o PSCAD 4.2.1 (ou PSCAD V4), versão instalada, é um aplicativo de 32 bits, ele é compatível apenas com versões 32 bits do MATLAB. Além disso, essa versão do PSCAD só consegue se comunicar até a versão MATLAB R2008b, e a versão instalada na máquina no momento era MATLAB R2009a.

Teoricamente, ao instalar qualquer versão anterior à 2009a na máquina, os problemas seriam resolvidos, porém ao tentar instalar versões anteriores, observaram-se muitos problemas de incompatibilidade com o sistema operacional instalado no computador em uso, o Windows 7. Observe na Tabela 2.1 um sumário dos problemas de compatibilidade encontrados. Isso mostrou que integrar os softwares utilizando o sistema operacional Windows 7 mostrar-se-ia um maior desafio.

**Tabela 2.1 - Compatibilidade Windows7/PSCAD V4 - MATLAB**

| | MATLAB<br>Versão < 2009a | MATLAB<br>Versão ≥ 2009a |
|---|---|---|
| PSCAD 4.2.1 | OK | Incompatível |
| Windows 7 | Problemas de compatibilidade | OK |
| Windows XP | OK | OK |

A fim de ganhar tempo, foi escolhido outro computador, munido do Windows XP, cujos problemas de compatibilidade são consideravelmente menores, para instalar todos os componentes necessários para a integração do sistema.

Durante o mesmo período, também foi tentado instalar a nova versão do PSCAD 4.3.1 (ou PSCAD 4X), também obtida através da comunicação com o suporte técnico. Mas, devido problemas obtidos como os listados a seguir, foi decidido seguir outra direção, focando sua utilização em Windows XP, uma vez que existe a possibilidade de se criar uma máquina virtual utilizando esse sistema operacional em uma máquina física que possui o sistema operacional Windows 7.

- Dificuldade em fazer o PSCAD X4 enxergar o MATLAB, algo que já havia sido resolvido na versão anterior, sendo então preciso mais comunicações com o suporte técnico;
- Incapacidade da nova versão de salvar arquivos na extensão '.psc', salvando-os somente em '.pscx', o que poderia trazer dificuldades ao se tentar acessar arquivos em outras máquinas que não possuem a nova versão.

Então, após todas as devidas instalações em um computador utilizando o Windows XP, foi encontrado mais um impasse. Após rápida comunicação com o suporte técnico, constatou-se que o sistema operacional estava instalado no driver "D", o que poderia estar causando problemas com alguns arquivos do PSCAD.

Tendo isso em vista, além de todos os problemas encontrados até então, uma decisão final foi tomada: instalar tudo aquilo requerido para a integração em um computador em que apenas o Windows XP estivesse instalado no driver "C", a fim de eliminar quaisquer conflitos existentes, além de todos os problemas encontrados até agora.

Para atingir tal objetivo sem que fosse necessária a formatação de nenhuma máquina, utilizou-se uma máquina virtual. A máquina virtual é um programa cuja finalidade é emular em um sistema físico em outros sistemas operacionais de maneira a fazê-los funcionar de modo semelhante a uma máquina física (MÁQUINASVIRTUAIS, 2008).

Utilizou-se o software VMWare para criar a máquina virtual. Nela, deve-se instalar um sistema operacional como se o estivesse instalando em uma máquina comum. Então, basta mandar o programa emular a máquina, como pode ser visto na Figura 2.2. Com isso, se tem uma máquina com as características desejadas para instalar os softwares. A máquina virtual é altamente customizável, mas não se entrará em detalhes de sua configuração, uma vez que este não é o escopo do trabalho.

VMware Player
VMware Player
vmware
Commands
Open
Browse for available virtual machines. When you select a virtual machine, it opens in this VMware Player window.
Download
Download a virtual appliance from VMware Virtual Appliance Marketplace. You can then open it in VMware Player.
Recent Virtual Machines
WinXP
Featured Virtual Appliance
FalconStor
Download
FalconStor Network Server Storage
FalconStor (NSS) is a pre-configured, production ready- virtual machine that's cost-effective, with rich virtual SAN solutions for SMB & remote sites.
VMware Player
VMware Player

**Figura 2.2 - VMWare Player - Emulação de Máquinas Virtuais**

Uma vez que o sistema era novo, tudo teve de ser instalado novamente. Desta vez foram instalados, nessa ordem:

- .NET Framework 4.0;
- Microsoft SDK;
- Microsoft Visual Studio 2008;
- PSCAD V4 (4.2.1);

- MATLAB R2008b;
- Intel Visual Fortran Compiler 11.1.067.

Vale observar que os três primeiros softwares instalados são, falando de maneira bastante simplificada, apenas bibliotecas de linguagens de programação da Microsoft, sendo elas necessárias para o funcionamento do sistema.

Com isso, o sistema passou a funcionar corretamente após algumas correções utilizando o aplicativo FortranMedic, mas sem a necessidade de se comunicar com o suporte técnico.

Posteriormente, retornou-se ao computador onde o sistema operacional estava instalado na partição "D", refazendo esse procedimento, obtendo assim êxito no funcionamento do sistema integrado.

Tendo finalmente obtido êxito na instalação e integração do sistema no Windows XP, retornou-se ao primeiro sistema operacional tentado, o Windows 7. O problema então encontrado foi a instalação de uma versão do MATLAB compatível com o PSCAD. Porém, após alguma pesquisa, encontrou-se a solução do problema (TECHNET, 2009), sendo necessária a atualização do JAVA, e a definição do Tema do Windows 7 para o Clássico, entretanto a simples atualização do Windows e do JAVA permitiram a instalação do MATLAB R2008b.

Após a instalação da versão compatível do MATLAB, R2008b, ao instalar o PSCAD 4.2.1, e então o compilador Intel Visual Fortran 11.1.067, a completa integração foi finalmente obtida.

## 2.4. Considerações Finais

A utilização integrada dos softwares PSCAD e MATLAB/SIMULINK pode vir a trazer muitos benefícios a simulações feitas para projetos da Universidade Federal do Ceará.

Essa integração, porém, não se deu de maneira simples e direta. Várias tentativas foram feitas, com diversos softwares diferentes, encontrando inúmeros obstáculos, pesquisando soluções e obtendo auxílio direto do suportes técnico, no intuito de se obter a completa integração.

Finalmente, após a aquisição do software recomendado pelo suporte técnico, além de muito trabalho, foi realizada a integração dos softwares. O sucesso dessa integração abrirá novas oportunidades para simulações não somente no projeto de simulação de microrredes, mas para vários projetos que possam vir a utilizar as capacidades combinadas de ambos os softwares.

# Capítulo 3

# Descrição e Instalação dos Componentes necessários para a completa integração entre os Softwares

## 3.1. Considerações iniciais

Após várias tentativas para integrar corretamente os softwares PSCAD e MATLAB, fazendo-os comunicar-se de maneira efetiva, finalmente se obteve sucesso nessa integração. Vários dos problemas que foram enfrentados durante o processo de instalação poderiam ter sido facilmente evitados caso houvesse algo como um roteiro simples, mostrando tudo que é necessário de maneira rápida e efetiva. Por exemplo, ao adquirir o Intel Visual Fortran, não foram encontrados documentos a ele relacionados explicitando que seriam necessários outros programas para o seu funcionamento, mesmo que gratuitos e facilmente encontrados, o que gerou um atraso considerável no processo de instalação do software.

O intuito deste capítulo é mostrar como instalar o software de maneira rápida e efetiva, evitando que uma possível dificuldade encontrada possa atrapalhar futuros usuários, já tendo sida resolvida neste trabalho.

Vale ressaltar que o que aqui está descrito foi apenas a maneira em que se obteve êxito na instalação e integração dos softwares em questão. Possivelmente esta não é a única maneira de se proceder. Ainda é possível que algum destes passos seja supérfluo, mas como se obteve êxito repetidas vezes seguindo este procedimento, e como o intuito do trabalho não gira somente na instalação do software, o procedimento será assim apresentado.

## 3.2. Processo de instalação de todos os componentes

Para instalar corretamente o compilador Intel Visual Fortran, deve-se primeiro verificar a compatibilidade dos softwares a serem instalados.

A versão adquirida do compilador Intel Visual Fortran foi 11.1.067, que é compatível com PSCAD 4.2.1, com o Microsoft Visual Studio 2008, além de ser compatível com as versões do MATLAB a partir da R2008a.

A versão utilizada do PSCAD foi 4.2.1, e é compatível com o compilador adquirido, e com as versões de MATLAB de R2006a a R2008b.Os problemas de compatibilidade entre PSCAD e MATLAB também podem ser observadas na Tabela 2.1.

Conhecendo todos os problemas de compatibilidade, os programas e suas versões foram escolhidos como segue:

- .NET Framework 4.0;
- Microsoft SDK;
- Microsoft Visual Studio 2008 SP1 (Service Pack 1);
- MATLAB R2008b;
- PSCAD V4 (4.2.1);
- Intel Visual Fortran compiler 11.1.067

Para a instalação aqui descrita, é suposto que a máquina em que se deseja instalar os softwares para comunicação não possui nenhum destes softwares instalados. Caso algum deles já esteja instalado, deve-se verificar a ordem de instalação, para que um programa possa observar que o outro já está instalado. Essa ordem de instalação é valida somente para os três softwares principais: MATLAB, PSCAD e o compilador, que devem ser instalados nesta ordem.

Primeiramente, deve-se instalar o .NET Framework 4.0, uma vez que ele é pré-requisito para instalação de todos os outros. Este está disponível gratuitamente na internet, sendo facilmente encontrado seu *webinstaller* no site da Microsoft.

Então, o Microsoft SDK deve ser instalado, cujo *webinstaller* também é facilmente encontrado no site da Microsoft. Neste ponto, foram instalados todos os componentes do Microsoft SDK, porém, pode-se dizer que os compiladores de C++ e os cabeçalhos e bibliotecas do Windows são os componentes de maior relevância.

Posteriormente, o Microsoft Visual Studio 2008 deve ser instalado. Utilizar a versão *webinstaller* obtida no site da Microsoft não mostrou bons resultados. Para instalar este software, foi adquirido o CD de instalação do Microsoft Visual Studio 2008 Express, a versão gratuita do software. Uma atualização dos componentes deste programa para a versão com o Service Pack 1 (SP1) mostrou-se necessária para o reconhecimento do software pelo compilador Intel Visual Fortran. Aqui, instalou-se somente o componente Microsoft Visual C++ 2008 Express Edition.

Até este ponto, a ordem de instalação não é realmente relevante, porém, a partir daqui, a instalação dos softwares deve seguir esta ordem, a fim de permitir que os softwares instalados posteriormente sejam capazes de reconhecer aqueles previamente instalados. Caso seja necessária a reinstalação de algum dos últimos após outros estarem instalados, deve-se reinstalar os outros softwares. É possível que simplesmente reparando a instalação dos softwares seja suficiente para o caso em questão.

Primeiramente, deve-se instalar o MATLAB, como recomendado pelo suporte técnico. A razão disso é que nem o PSCAD nem o compilador reconhecerão esta instalação caso isso não seja seguido. É bom ressaltar neste ponto que o MATLAB aqui a ser instalado deve ter sua versão em 32 bits, uma vez que a versão utilizada do PSCAD é um software de 32 bits, sendo assim incompatível softwares de 64 bits (ver apêndice A). Fora esse detalhe, sua instalação deve se dar normalmente, sem nenhuma outra configuração específica para a integração.

O PSCAD, que deve ser instalado a seguir, não necessita de qualquer configuração específica, i.e., pode ser instalado normalmente para a futura integração.

Já o compilador Intel Visual Fortran deve ser instalado por último, para que este registre todas as variáveis de ambiente e localizações dos softwares já instalados.

Seguindo estes passos, os softwares devem ser capazes de se reconhecer. Para verificar isso, observe as instruções dadas no início da próxima seção.

Foi desenvolvido um fluxograma na Figura 3.1 contendo os passos para a ordem de instalação dos softwares.

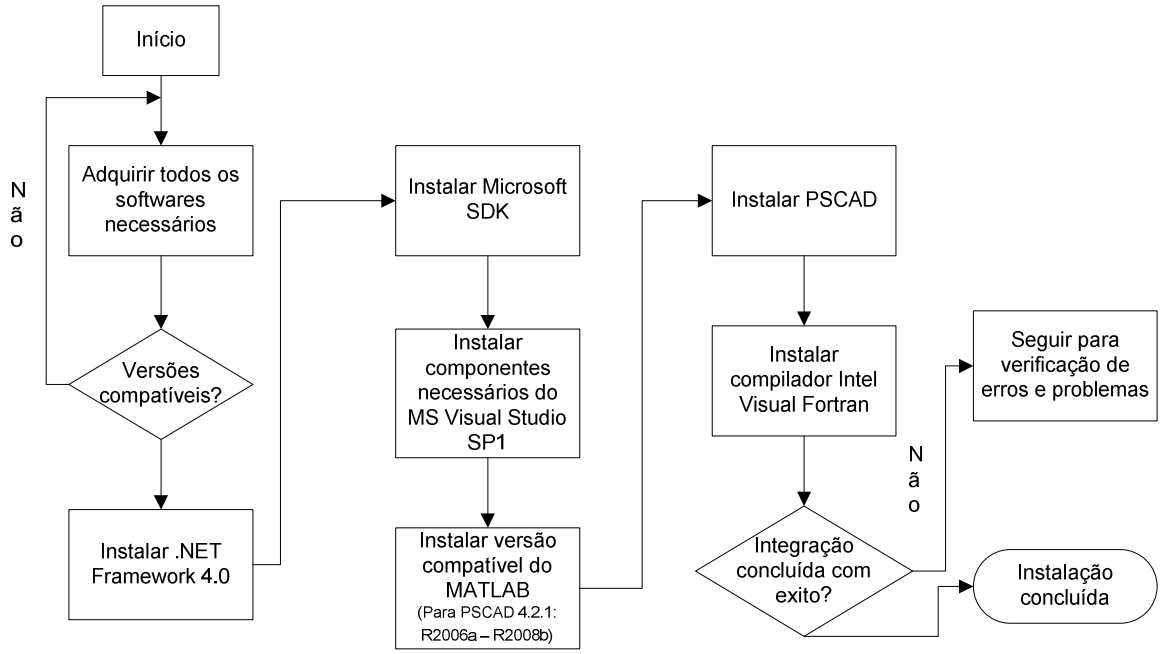



**Figura 3.1 - Fluxograma – Instalação**

## 3.3. Configurações e Solução de Problemas

Ao instalar corretamente o novo compilador, o PSCAD deve ser capaz de reconhecê-lo, habilitando assim sua escolha no menu.

Depois de concluídos todos os passos descritos anteriormente, o software PSCAD deve ser capaz de utilizar o novo compilador instalado em qualquer simulação. Originalmente, o PSCAD vem configurado para utilizar um compilador gratuito, o GNU Fortran, que é capaz de gerar simulações mais simples.

É preciso então configurar o PSCAD para utilizar o novo compilador. Para isso deve-se acessar o menu “Edit” da barra de ferramentas, e escolher a opção “Workspace Settings”, como na Figura 3.2. Na aba “Fortran”, deve-se selecionar a opção “Intel(R) Visual Fortran 11”, em “Installed Version”, como pode ser visto na Figura 3.3 e então pressionar OK

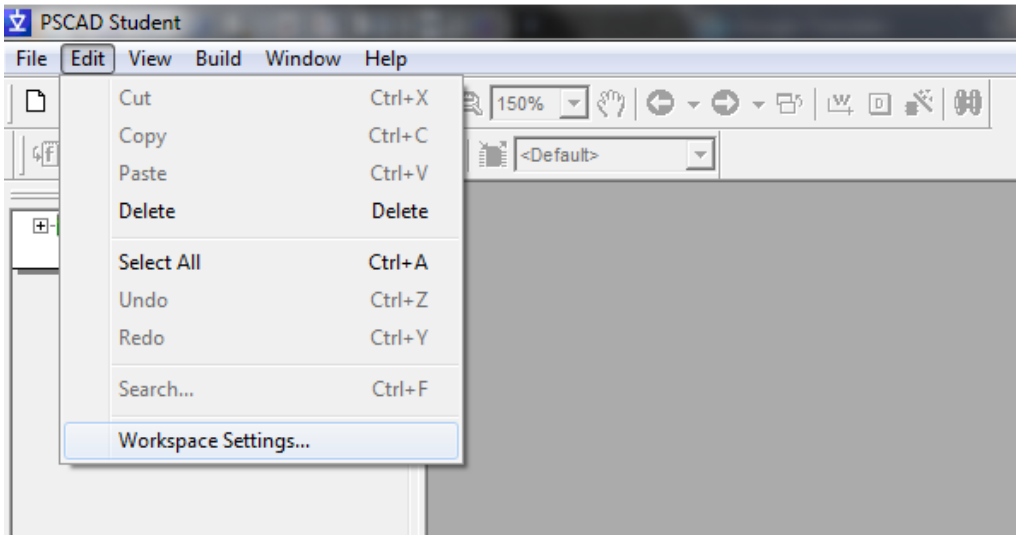



**Figura 3.2 - Localização Workspace Settings**

Caso esta opção não esteja visível, é possível que algo tenha saído errado. O principal problema enfrentado desde o início do trabalho está nesse ponto. Após a instalação do PSCAD e do compilador Intel Visual Fortran, o PSCAD não reconhecia o compilador.

Como já foi explicitado, após contato com o suporte técnico com o fim de corrigir esse problema, e de algumas tentativas sem sucesso, foi obtido um aplicativo que verifica as conexões do compilador com o PSCAD, o FortranMedic (Figura 2.1). Vale citar que este aplicativo foi aprimorado pelo próprio suporte técnico a fim de suprir as necessidades que foram encontradas no decorrer da instalação.

Ao abrir este aplicativo, observam-se em sua tela inicial as instruções de uso. No menu “Actions”, seleciona-se a opção “Start”, para que tenha início o processo de verificação. Esse processo pode demorar algum tempo. Terminada a verificação, será exibido no aplicativo o relatório gerado contendo as informações sobre os softwares a serem utilizados. Quaisquer linhas em vermelho significam algum problema encontrado, mas nem toda linha vermelha significa o mau funcionamento da integração entre ambos os softwares. Serão citados aqui alguns pontos que podem levar a tal problema.

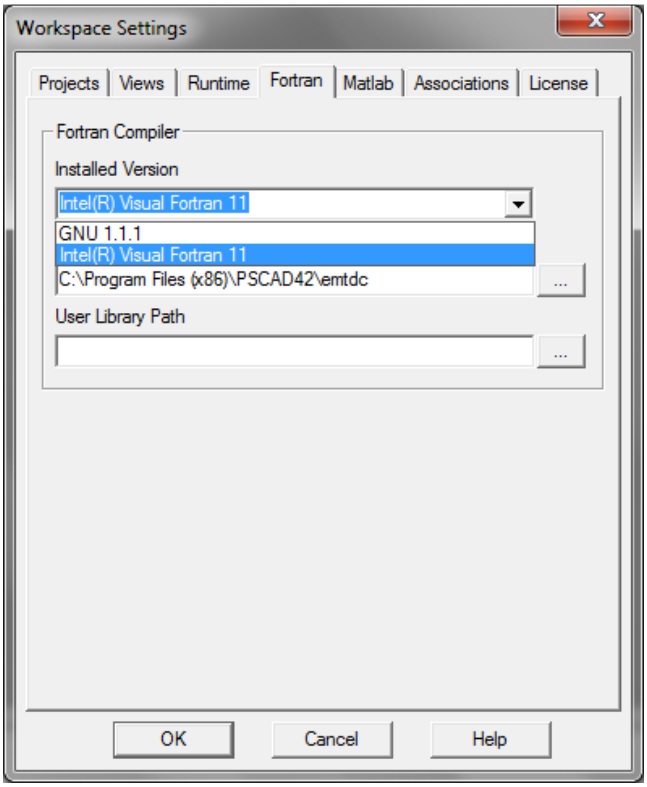



**Figura 3.3 – Workspace Settings - Compilador em uso**

### 3.3.1. Verificação das variáveis de ambiente

Primeiro, pode-se observar os pontos citados na Seção 2.3 relacionados às variáveis de ambiente. Nesse caso, é possível que algum outro programa esteja causando um conflito com as variáveis de ambiente geradas pelo novo compilador. Esses pontos podem vir a corrigir o problema obtido.

Caso essas variáveis não sejam observadas em 'PATH Environment Variable' e em 'LIB Environment Variable', seções do relatório gerado pelo FortanMedic, não significa que as variáveis estão erradas e a interface não está funcionando por isso, mas apenas que as variáveis foram configuradas de maneira diferente. Infelizmente não é possível conhecer todos os pontos nesse caso, e é aconselhável entrar em contato com o suporte técnico do PSCAD, enviando o relatório salvo no aplicativo caso nenhum dos próximos pontos resolva o problema, em *support@pscad.com*.

### 3.3.2. Verificar reconhecimento do PSCAD

Deve-se então verificar se o PSCAD foi devidamente reconhecido. Isso pode ser verificado na seção “Installed PSCAD Versions”, como pode ser observado na Figura 3.4.

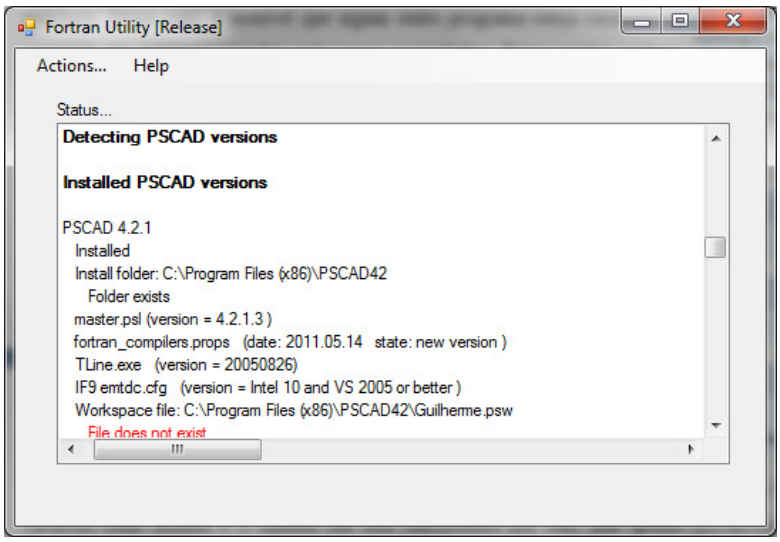



**Figura 3.4 - FortranMedic - Versões do PSCAD Instaladas**

Caso o PSCAD não seja reconhecido, deve-se reinstalá-lo, e então o compilador novamente.

### 3.3.3. Verificação da instalação do compilador

Se o PSCAD estiver devidamente instalado, deve-se então verificar a instalação do compilador. Para isso, procura-se a seção “Intel Fortran (11.1.067)”, como pode ser visto na Figura 3.5, a fim de verificar se há alguma linha vermelha nesse ponto. Caso haja linhas vermelhas acusando apenas falta de registro na subseção “Environment (Required)”, embora exista a pasta, é possível que, ao clicar com o botão direito do mouse nessas linhas, corrigir-se-á o erro, adicionando o que for necessário ao registro, e uma vez que os arquivos estão todos no local correto, o problema seja corrigido.

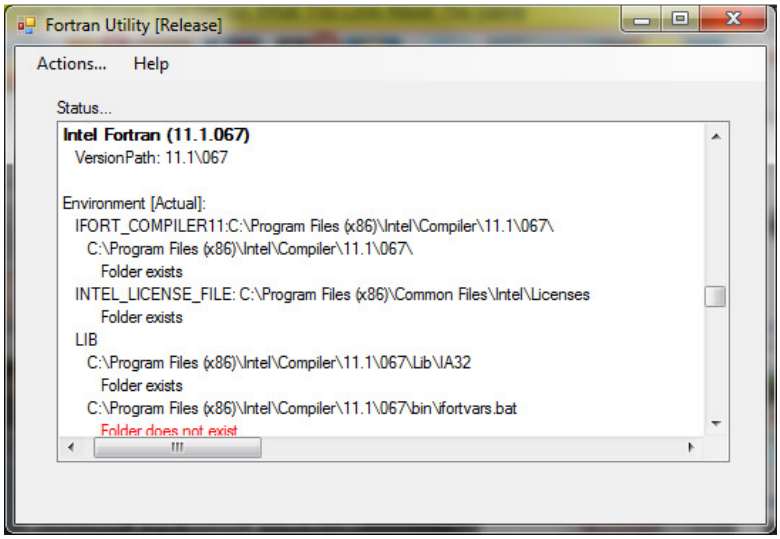



**Figura 3.5 - FortranMedic - Verificação do compilador**

Se as linhas em vermelho acusarem a falta das pastas, então será necessário reinstalar o compilador, repetindo essa verificação após a reinstalação.

### 3.3.4. Verificação do arquivo Fortran_compilers.props

Talvez seja preciso atualizar a versão de um arquivo, o Fortran_compilers.props, como visto na Figura 3.6. Para corrigir este problema, pressione o botão direito do mouse em cima da linha que aponta o erro. Esse erro deixa claro que o compilador não será reconhecido caso este arquivo não seja atualizado. Se essa correção permitir que a opção do novo compilador seja selecionada no PSCAD, tente simular um sistema simples. Caso a simulação apresente um erro, tente reiniciar o computador e então simular novamente. Caso o erro persista, siga para os pontos a seguir.

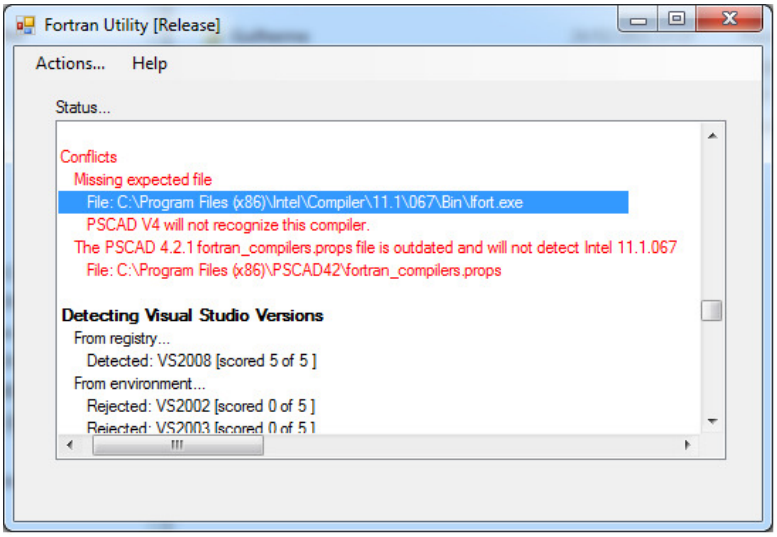

Fortran Utility [Release]
Actions... Help
Status...
Conflicts
Missing expected file
File: C:\Program Files (x86)\Intel\Compiler\11.1\067\Bin\Ifort.exe
PSCAD V4 will not recognize this compiler.
The PSCAD 4.2.1 fortran_compilers.props file is outdated and will not detect Intel 11.1.067
File: C:\Program Files (x86)\PSCAD42\fortran_compilers.props
Detecting Visual Studio Versions
From registry...
Detected: VS2008 [scored 5 of 5 ]
From environment...
Rejected: VS2002 [scored 0 of 5 ]

**Figura 3.6 - FortranMedic - Verificação do fortran_compiler.props**

### 3.3.5. Verificação da versão do Microsoft Visual Studio

O próximo ponto a ser observado será a detecção das versões instaladas do Visual Studio. Para isso, avance até a seção "Detecting Visual Studio Versions" e verifique qual a "pontuação" do "VS2008". Se esta estiver 5 de 5, além de se encontrar na subseção "From registry", como pode ser visto na Figura 3.7, então o Visual Studio 2008 está corretamente instalado.

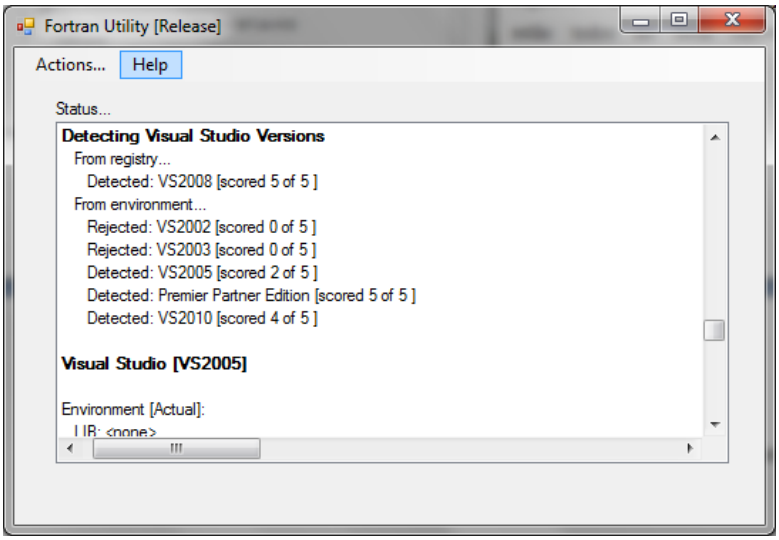

Fortran Utility [Release]
Actions... Help
Status...
Detecting Visual Studio Versions
From registry...
Detected: VS2008 [scored 5 of 5 ]
From environment...
Rejected: VS2002 [scored 0 of 5 ]
Rejected: VS2003 [scored 0 of 5 ]
Detected: VS2005 [scored 2 of 5 ]
Detected: Premier Partner Edition [scored 5 of 5 ]
Detected: VS2010 [scored 4 of 5 ]
Visual Studio [VS2005]
Environment [Actual]:

**Figura 3.7 - FortranMedic - Versões do Visual Studio**

É comum acontecer de a pontuação do “VS2008” seja 4 enquanto a pontuação do “Premier Partner Edition” é 5, colocando este último na subseção “From registry”. Esta configuração não se mostra suficiente para que o PSCAD exiba a opção do novo compilador. Se isso acontecer, tente atualizar a o Microsoft Visual Studio. Vale lembrar também que a versão utilizada do compilador não é compatível com o Microsoft Visual 2010.

### 3.3.6. Contato com Suporte Técnico

Após todos estes pontos, caso a instalação não tivesse tido sucesso na primeira vez, ela deve ter sido bem sucedida agora, habilitando assim a opção “Intel® Visual Fortran” na escolha do compilador no PSCAD.

Caso não se tenha obtido sucesso na integração a seguir estes pontos, sugere-se entrar em contato com o suporte técnico do PSCAD em support@pscad.com.

Para certificar o funcionamento do compilador, é bom utilizá-lo em uma simulação simples, que pode ser desenvolvida pelo usuário ou então carregada a partir dos exemplos de simulação disponibilizados pelo PSCAD.

### 3.3.7. Configurações de integração.

Tendo então finalmente instalado o compilador junto ao PSCAD, deve-se ajustar as configurações do MATLAB, para que o PSCAD reconheça as suas bibliotecas de integração. Para isso, deve-se ir novamente acessar o menu “Workspace settings” (Figura 3.2), seguir à aba MATLAB, como pode ser visto na Figura 3.8. Aqui, deve ser escolhida como versão instalada a Versão 5, a fim de ser possível definir a localização das bibliotecas do MATLAB.

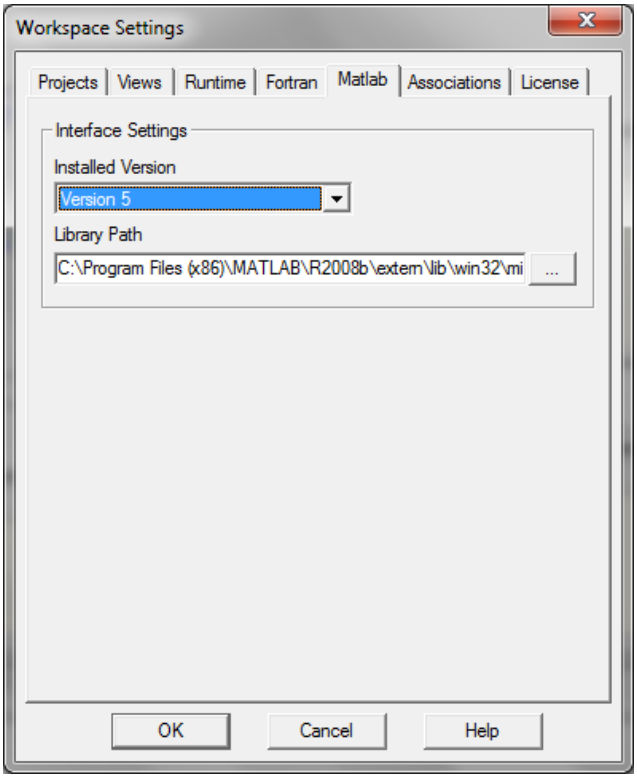



**Figura 3.8 - Workspace Settings - MATLAB**

As bibliotecas devem estar localizadas na pasta onde se encontra instalado o MATLAB. Por exemplo: se o MATLAB está instalado, como mostra a figura, na pasta "*C:\Program Files (x86)\ MATLAB\2008b*", as bibliotecas se encontrarão em: "*C:\Program Files (x86)\ MATLAB\2008b\extern\lib\win32\microsoft*".

Dessa maneira, nomeando a pasta onde o MATLAB foi instalado de *"%MATLABROOT%",* o caminho completo para as bibliotecas do MATLAB que deverá ser definido em "Library Path" será (ver ANEXO A):

- *%MATLABROOT%\extern\lib\win32\microsoft*

Tendo definido as bibliotecas do MATLAB e o compilador a ser utilizado no PSCAD, o PSCAD deverá finalmente ser capaz de se comunicar devidamente com o MATLAB.

A fim de verificar essa comunicação, pode-se utilizar, mais uma vez, dos exemplos relacionados ao MATLAB, também disponibilizados pelo PSCAD, que se encontram na pasta de exemplos, dentro da pasta de instalação do PSCAD, como também explicitado no ANEXO A.

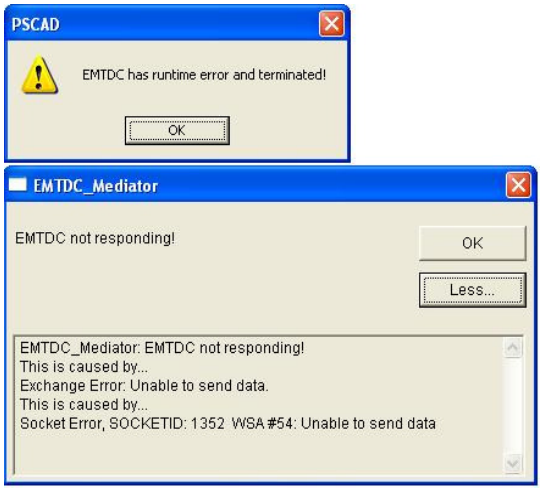



**Figura 3.9 - Erro gerado pela falta do M-File**

Vale lembrar também que o M-File, salvo pelo MATLAB deve estar incluso na pasta especificada pelo componente do MATLAB presente no PSCAD. Caso contrário, a simulação apresentará um erro como o mostrado na Figura 3.9. Esse erro ocorre porque o PSCAD não é capaz de encontrar o M-File referente à simulação, impossibilitando assim a comunicação entre os softwares. Mais informações sobre como localizar o arquivo do MATLAB serão apresentadas no Capítulo 4.

## 3.4. Considerações Finais

Inicialmente a instalação dos softwares se deu no Windows XP. Posteriormente, feitas as devidas modificações, foi possível instalar as versões de MATLAB compatíveis com o PSCAD no Windows 7. A instalação dada no sistema operacional mais antigo apresentou muito menos problemas, sendo até mesmo possível que a verificação de erros utilizando o aplicativo FortranMedic seja desnecessária, enquanto que utilizando o sistema operacional mais atualizado, mais erros foram encontrados. Porém, mesmo com a dificuldade superior, ao utilizar os métodos aqui apontados, a instalação foi concluída com sucesso. Em ambos os sistemas operacionais se obteve êxito na instalação e completa integração da interface PSCAD – MATLAB.

A integração dos softwares foi finalmente obtida com sucesso. O intuito deste trabalho em relação à instalação do software foi concluído, expondo maneiras de se obter êxito na integração dos softwares em questão.

Foi desenvolvido um fluxograma na Figura 3.10 para melhor visualização dos passos aqui descritos sobre problemas de integração.

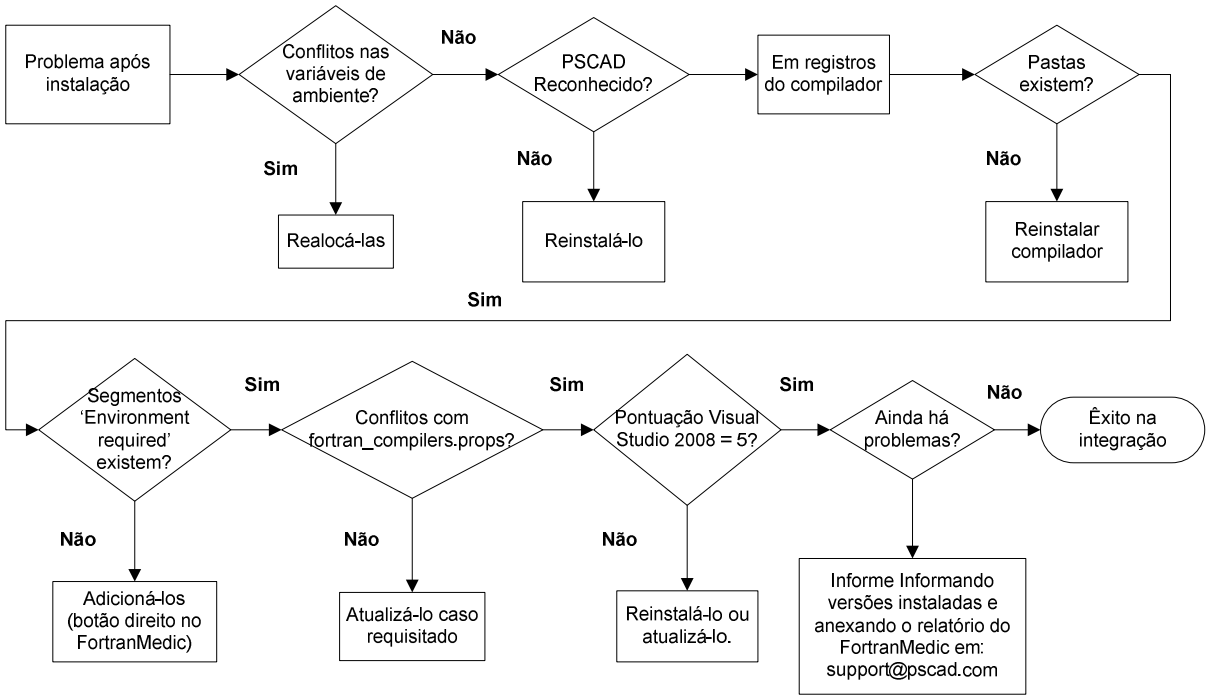



**Figura 3.10 - Fluxograma - Solução de erros**

# Capítulo 4

# Ferramentas e Métodos de Integração entre os Softwares

## 4.1. Considerações Iniciais

Após tudo ser devidamente instalado, tendo um sistema que permite a comunicação entre os softwares, é preciso definir como a mesma é realizada e quais são os parâmetros utilizados por cada um deles, a fim de que se tenha total harmonia em ambos os lados da simulação. Para isso, são necessários conhecimentos em MATLAB e em PSCAD, focando em Fortran, linguagem de programação utilizada no PSCAD.

A Figura 4.1 mostra de maneira simplificada como ocorre a comunicação entre os softwares, o que pode ajudar na compreensão do funcionamento da interface. O bloco DSDYN é o responsável pela execução de segmentos em Fortran gerados pelo PSCAD.

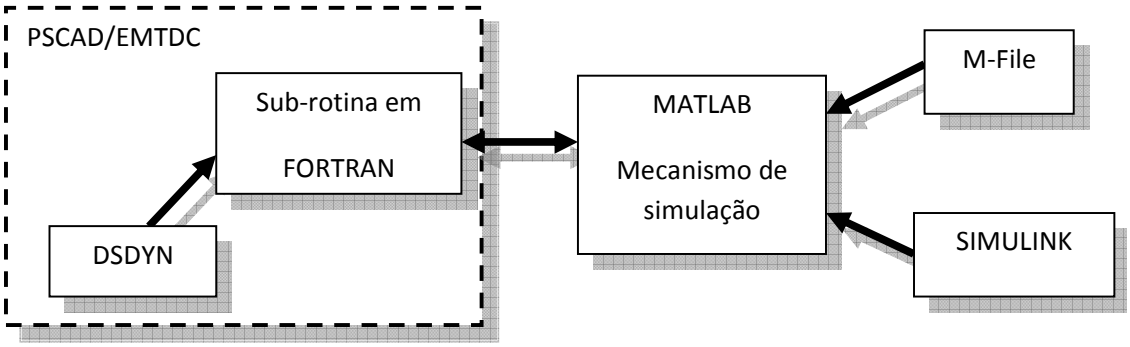



Figura 4.1 - Funcionamento da Interface

Uma vez que em um ambiente de engenharia conhecimentos em MATLAB são mais comuns que em PSCAD, ou em Fortran, será dado um maior foco às configurações no PSCAD, apresentando maiores detalhes sobre este software, além de mais ferramentas que poderão ser utilizadas no desenvolvimento das simulações.

A intenção de apresentar mais ferramentas disponíveis no PSCAD é de reduzir o esforço computacional necessário no MATLAB, reduzindo assim também a quantidade de dados trocados, visando a simplificação e o aumento de velocidade da simulação.

## 4.2. Utilização de arquivos do MATLAB

A comunicação com o PSCAD pode se dar através de dois dos métodos de simulação do MATLAB: o M-File e o SIMULINK. Em ambos os casos, é necessário que sejam definidos quais os parâmetros que serão recebidos do PSCAD e enviados para o PSCAD, a fim de harmonizar a comunicação.

A comunicação feita através de M-Files, arquivos simples e programáveis do MATLAB, é feita pela definição de uma função. O M-File deve ser desenvolvido como uma função do MATLAB, onde devem ser definidas as entradas e as saídas dessa função. Essas entradas e saídas serão os parâmetros definidos para a comunicação com o PSCAD, onde se dará a troca de dados.

Já no SIMULINK, blocos, ou componentes, de troca de dados terão esse papel de realizar a comunicação com o PSCAD. Blocos de entrada de dados receberão dados vindos do PSCAD e blocos de saída de dados enviarão dados para o PSCAD.

Mais informações sobre como configurar estes arquivos serão apresentadas ao longo do capítulo.

## 4.3. PSCAD

Para que haja uma correta troca de dados entre o PSCAD e o MATLAB, é necessário que em ambas as partes da simulação sejam definidos os dados a serem enviados e recebidos. Deve haver congruência de dados, a fim de harmonizar a simulação, fazendo com que ambas trabalhem em conjunto, como se espera.

O PSCAD oferece várias ferramentas para simulação. Estas devem ser conhecidas, uma vez que utilizar apenas um software para simulação é bem mais simples e cômodo, e a utilização de uma simuluação conjunta deve ser evitado se possível.

Para que um arquivo de simulação no PSCAD consiga se comunicar corretamente com o MATLAB, é necessário habilitar a comunicação da simulação com as bibliotecas do MATLAB. Para se fazer isso, basta entrar nas configurações do projeto, pressionando o botão direito em qualquer local vazio da área de projeto, seguir até a aba "Link" e selecionar a opção "Link this simulation with the currently installed Matlab libraries", como pode ser visto na Figura 4.2.

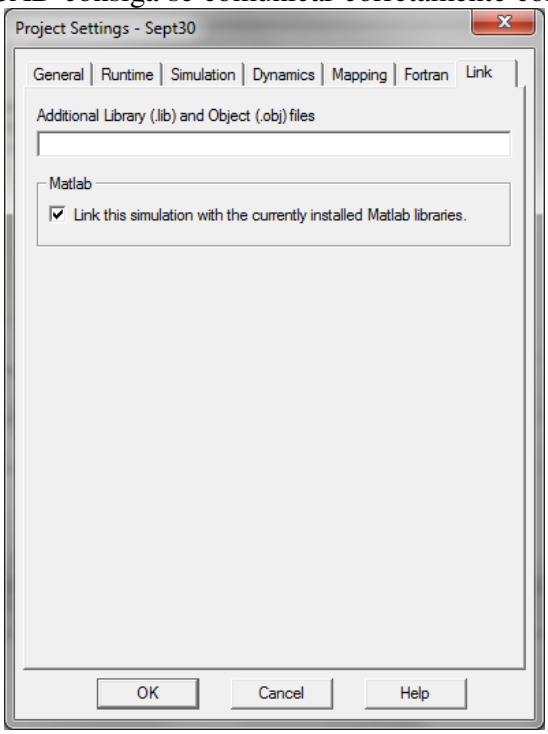



**Figura 4.2 - Link projeto - bibliotecas MATLAB**

### 4.3.1. Blocos de Componentes

Inicialmente, deve-se conhecer a função do PSCAD de criar novos blocos, que podem exercer inúmeras funções, dependendo apenas do que neles for programado.

Estes novos blocos podem ser utilizados para reduzir o tamanho de circuitos, simplificar sistemas construídos ou fazer o tratamento de dados neles. Além disso, vários novos blocos podem compor uma nova biblioteca, fornecendo assim mais opções para utilização no desenvolvimento de sistemas elétricos. Lembrando que para a utilização dessa nova biblioteca é necessário que a mesma esteja carregada no "Workspace" do PSCAD.

Nestes blocos, podem ser definidas entradas e saídas, de maneira a suprir sua função. Estas podem ser de dados (entrada ou saída) ou elétricas, fazendo parte de algum sistema elétrico construído.

Estes blocos são configuráveis, de modo que pode ser escolhido criar uma página dentro deste e então montar novo sistema utilizando ferramentas já existentes, ou simplesmente utilizar-se de programação para definir a função deste novo bloco. Para criar um novo bloco, deve-se

selecionar a ferramenta “New Component”, localizada na barra de atalhos do PSCAD, como pode ser visto na Figura 4.3.

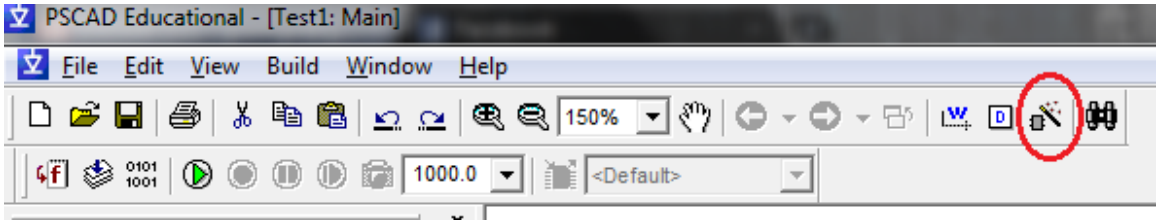



**Figura 4.3 - New Component - Criar um novo componente**

Após pressionar o botão de criar novos componentes, uma nova janela será aberta, como visto na Figura 4.4, a fim de se configurar esse novo componente. Nessa tela serão definidos o nome do novo componente, o texto que será exibido no bloco, o número de “pinos” (que poderão ser de entrada ou saída de dados, ou de componentes elétricos) em cada lado do novo bloco e se haverá uma nova página para definição do novo componente.

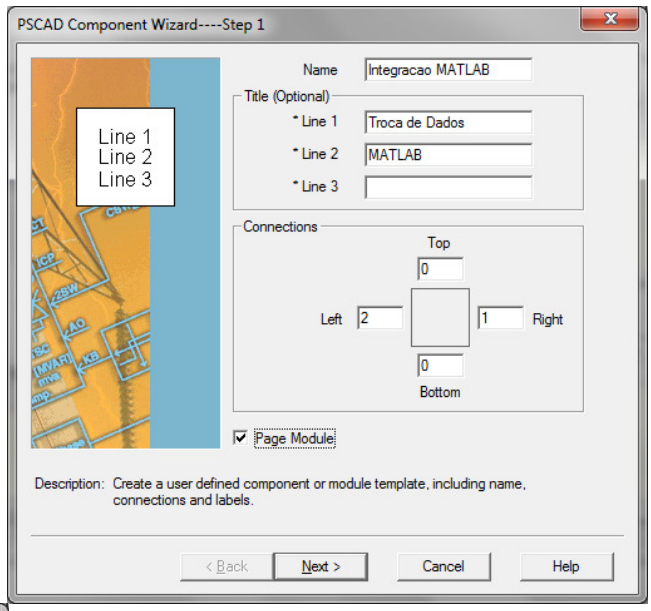



**Figura 4.4 - Criação de novo componente**

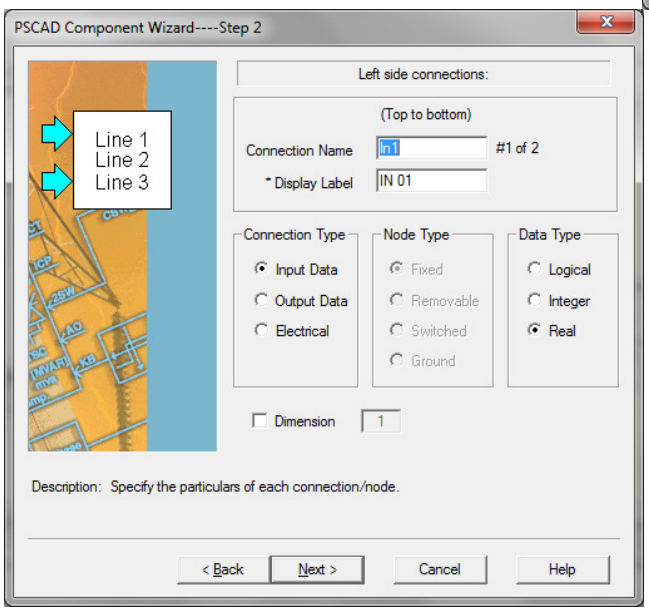



**Figura 4.5 - Definição da função de pinos**

Caso haja algum pino, as próximas telas trarão opções para a definição da função de cada um destes pinos, como pode ser visto na Figura 4.5, definindo o tipo de pino que este será, o tipo de dados ou de conexão elétrica dele, além da dimensão dos dados nele, caso deseje-se passar um vetor de dados por ele.

Por fim, caso tenha sido definido que este bloco possuiria uma página, deverão ser definidos os parâmetros dessa página.

Tendo concluído as definições de um novo componente, deve-se agora definir suas funções.

No caso de um componente que possui uma página para construção, este passo poderá ser mais simples, bastando apenas criar nessa página o sistema desejado, conectando pinos a eles. O bloco então se comportará, falando de maneira simples, apenas substituindo o sistema interno, construído na página, pelo bloco quadrado, economizando assim espaço. Se for necessário utilizar várias vezes este bloco, seria uma boa idéia construí-lo em uma biblioteca.

Caso não seescolha a opção página para construção de sistemas a parte, o bloco ficará na função das ferramentas disponíveis para programação.

### 4.3.2. Parâmetros

Se o bloco que está sendo construído for configurado para não possuir uma página interna a ele, algumas novas opções serão habilitadas, apresentando funções interessantes, ampliando assim a capacidade destes novos componentes. Um exemplo destas opções seria a utilização de *Parâmetros*.

Pode-se configurar o bloco de maneira que ao pressionar duas vezes com o mouse em cima dele, alguns parâmetros editáveis se apresentarão ao usuário, assim como uma boa parte dos blocos disponíveis na biblioteca principal do PSCAD, aprimorando assim a capacidade de interação e configuração do bloco com o usuário.

Para criar essas configurações, deve-se defini-las nos parâmetros do bloco. E para isso, devem-se acessar suas definições, clicando com o botão direito no bloco e selecionando a opção “Edit Definitions”. Isso abrirá o modo de edição de componentes.

Antes de prosseguir, apenas para informação, para retornar à pagina principal do sistema, onde se encontra o novo componente que está sendo editado, deve-se apertar o botão indicado na **Erro! Fonte de referência não encontrada.**.

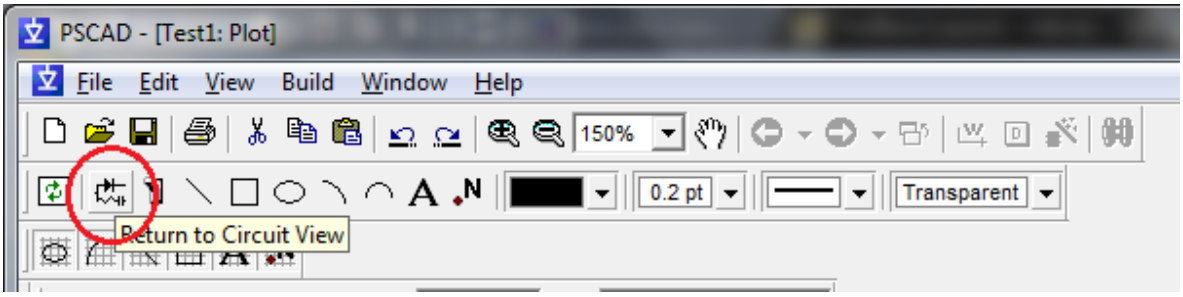



**Figura 4.6 - Retornar ao modo de construção de circuito**

Ao abrir o modo de edição de definições, podem ser observadas três abas: “Graphics”, “Parameters” e “Script”.

Na aba “Graphics” é definida a aparência do componente que está para ser redefinido, sendo possível utilizar diversas ferramentas de desenho. Neste ponto, também podem ser definidos novos pontos de conexão, clicando no botão “New Connection”. É preciso configurar essa nova conexão, definindo-a como entrada ou saída de dados ou se faz parte de um sistema elétrico (ver Figura 4.7). As configurações se mostram semelhantes àquelas descritas ao se criar um novo bloco.

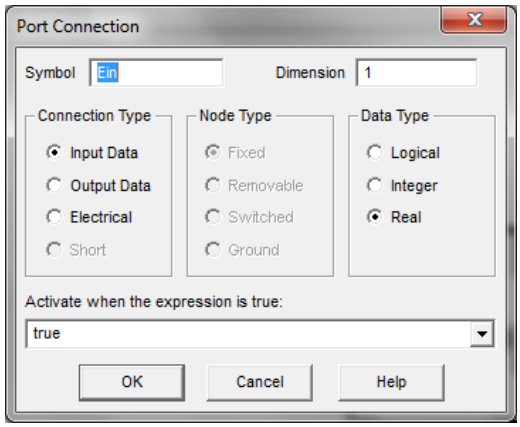



**Figura 4.7 - Configurações de porta de conexão**

Um parâmetro que pode ser utilizado é a função de ativação condicional. É possível criar uma condição para que o ponto de conexão seja ativado, baseando-se em algum parâmetro ou variável criada.

Na próxima aba, “Parameters”, poderão ser definidos os parâmetros para configuração, definindo variáveis a fim de configurar o funcionamento do sistema. Para isso, é preciso criar uma nova categoria, no botão “New Category”, no menu de atalhos, ao lado da caixa de texto. Criada a categoria, é possível adicionar a ela sua interface gráfica, adicionando “Text Labels”, ou caixas de texto, “Input Fields” ou campos de entrada, além de se poder criar “Choice Boxes” ou caixas de escolha a fim de aumentar ou simplificar as possibilidades de configurações. Utilizar o “Segment Manager” ajuda a organizar as categorias definidas.

As variáveis da nova categoria serão definidas nas ferramentas “Input Field” e “Choice Box”.

Ao inserir um “Input Field”, será adicionada à interface gráfica uma caixa de texto onde será possível entrar com escrita do teclado. Enquanto editando o componente, dar um duplo clique em um destes campos, se observará uma janela semelhante à da Figura 4.8, onde os campos devem ser devidamente definidos. O campo “Symbol” define o nome da variável a ser modificada pelo campo a ser configurado. Para acessar essa variável novamente nesse componente será necessário apenas chamar o texto aqui escrito, precedido pelo caractere “$”. No caso dessa figura, “$mfile”.

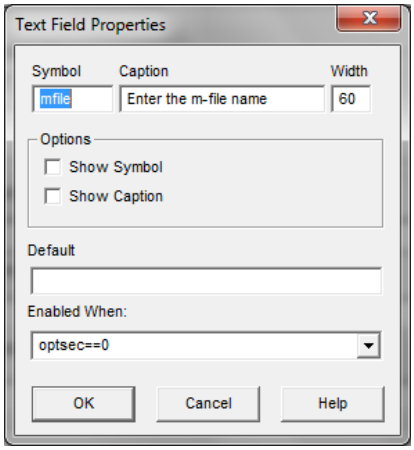



**Figura 4.8 - Parameters - Configuração de campo de entrada**

O campo “Caption” apenas define o texto escrito ao lado da caixa de entrada de texto da ferramenta escolhida. O campo “Width” limita o número de caracteres que podem ser adicionados no campo da nova variável. Com isso uma nova variável é definida.

No caso das ferramentas de “Choice Box”, ao editá-las, observa-se uma janela semelhante à da Figura 4.9. Nesta janela, ao pressionar o botão “Dup” se adicionam campos para a lista. O valor adicionado à lista é aquele escrito acima dela. Então se utilizam os outros botões para organizar as opções. Caso uma das opções oferecidas seja selecionada, o número designado a esta opção será armazenado na variável definida no campo “Symbol” desta janela.

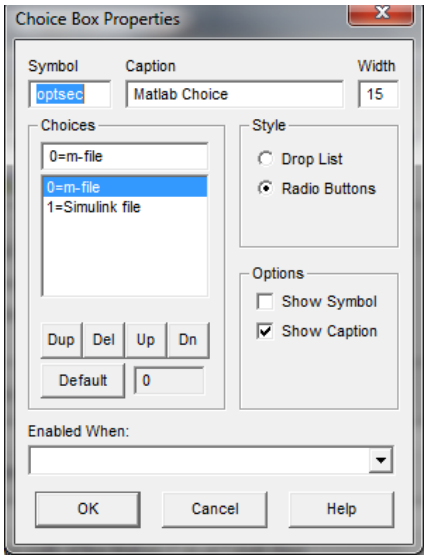



**Figura 4.9 - Parameters - Configuração de caixa de escolha**

É possível observar um exemplo de condição de ativação na Figura 4.8, que será ativada somente se o valor variável definida na Figura 4.9 for igual a zero, ou seja, seja selecionado a opção “m-file”.

Dessa maneira, ao finalizar a configuração de uma

categoria, podem-se criar novas categorias. As variáveis devem sempre ser diferentes, a fim de se evitar conflitos.

Para selecionar outras categorias utilizadas, basta selecionar a desejada nas opções que serão dadas ao dar um duplo clique no novo componente.

### 4.3.3. Script

Na próxima aba, nomeada "Script" existe uma série de opções e ferramentas que podem ser utilizadas para uma melhor configuração de dados, podendo utilizar ou definir variáveis, e manipulá-las da maneira que se desejar.

A integração com o MATLAB é feita nesta aba, a partir da programação feita utilizando a linguagem Fortran, no segmento Fortran.

Diferentes tipos de segmentos compõem o script, de modo que cada segmento faz parte dele, como se pode deduzir a partir do seu nome. Para habilitar os segmentos a serem utilizados, deve-se acessar o "Segment Manager" (Figura 4.10) na barra de atalho quanto à aba "Script" estiver aberta.

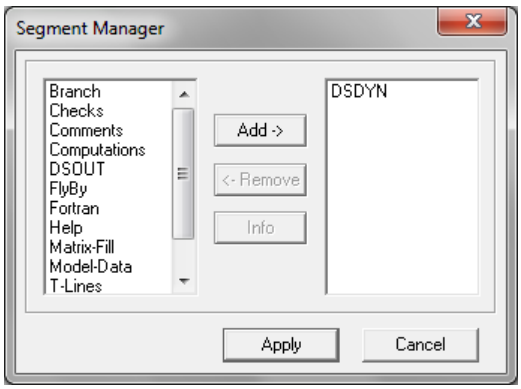



**Figura 4.10 - Segment Manager**

Alguns segmentos serão brevemente comentados a seguir. Maiores informações estão disponíveis em (MANITOBA, 2005).

#### 4.3.3.1 Computations

Algumas vezes pode ser necessária a utilização de algum artifício matemático antes do tratamento efetivo dos dados. Um exemplo disso é o caso de existir um campo de entrada de frequência em hertz, mas em questões matemáticas, deve-se utilizar seus valores em radianos.

Para casos como esse o segmento "Computations" oferece a capacidade de dar esse tratamento matemático nos dados antes de qualquer outra coisa, uma vez que é o primeiro

segmento a ser considerado. Quaisquer outros segmentos utilizados estarão usando as variáveis após o tratamento nesse segmento.

A capacidade de utilizar ferramentas matemáticas e operações lógicas podem ajudar a simplificar o sistema utilizado, como no exemplo citado em que ao receber do usuário a frequência em hertz simplifica a entrada de dados.

Para utilizá-lo, deve-se seguir o seguinte padrão:

<DataType> <Name> = <Expression>

Que pode ser exemplificado:

REAL VaPU = Va / Vbase

Neste segmento, a utilização do caractere "$" que define variáveis é desnecessária, de modo que sua utilização acarretará em erros.

#### 4.3.3.2 Branch

Este segmento é direcionado ao tratamento de informações elétricas, sendo utilizado em casos em que há entrada de grandezas elétricas.

Ele é definido basicamente pela definição do tipo e dos valores de componentes passivos, como resistores, indutores e capacitores, e a que pontos eles estarão conectados.

Para utilizá-lo, deve-se seguir este padrão:

[<Nome> = ] $<PARA> $<DE> [<Palavra chave>] [$]<R> [$][<L>] [$][<C>]

*tudo que se encontra entre colchetes ([ ]) são opcionais.

Onde:

- <DE> e <PARA> são os nomes definidos como "Symbol" das conexões (nós elétricos) definidas na aba "Graphics";
- <R> Valor da resistência entre os pontos em ohms;
- <L> Valor da indutância entre os pontos em henry;
- <C> Valor da capacitância entre os pontos em micro farad;

- <Branchname> = pode ser usado para nomear a ramificação a ser utilizada;
- <Palavra Chave> pode ser 'SOURCE' ou 'BREAKER', definindo uma fonte em série com o ramo em questão ou um ramo que muda.

Para se conectar ramos em paralelo, basta chamar uma segunda vez um ramo entre dois pontos já usados. Dessa maneira, esse novo ramo ficará em paralelo com o primeiro.

Por exemplo, para se obter entre os pontos N1 e N2, um circuito equivalente ao da Figura 4.11, deve-se entrar o seguinte código:

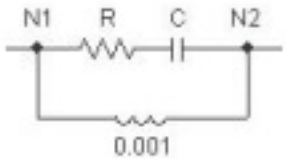



**Figura 4.11 - Exemplo de utilização de Branch**

```
$N1   $N2    $R   0.0   $C
$N1   $N2   0.0   0.001   0.0
```

#### 4.3.3.3 Fortran

Nesse segmento, qualquer código utilizando a linguagem de programação Fortran pode ser utilizado. O código deve estar formatado com o padrão Fortran 77 ou no padrão de script do PSCAD. É importante lembrar que para qualquer linha de comando nesse script deve ser precedido por seis espaços em branco.

É nesse segmento em que será desenvolvido o código necessário para integração do PSCAD com o MATLAB.

Uma vez que a programação para a integração é feita em Fortran, é possível se utilizar de artifícios de lógica de programação para dar suporte a essa integração, podendo aprimorar sua utilização.

Os apêndices B e D e o anexo B trazem exemplos dessa programação e serão devidamente explicados posteriormente.

## 4.4. Integração PSCAD – M-File

Uma das opções de integração do PSCAD com o MATLAB é utilizando um arquivo conhecido por m-file. Esse arquivo comporta apenas programação utilizando as funções do

MATLAB, e não se utiliza, de maneira direta, de bloco de simulação como o SIMULINK, por exemplo.

Para que seja efetuada a comunicação entre os dois softwares, ambos devem ser programados de maneira coerente e coordenada, a fim de que os dados enviados sejam interpretados corretamente, fazendo assim o tratamento correto destes dados.

Por isso, é preciso ter conhecimentos específicos em ambos os softwares. Segue uma breve descrição do que se deve ser feito para integrar corretamente os dados nos dois softwares, aplicada a exemplos, o que facilita a compreensão dos procedimentos seguidos.

O exemplo, uma aplicação simples, desenvolvida apenas para fins ilustrativos do método de integração, baseia-se um sistema RLC simples. O circuito simulado pode ser observado na Figura 4.12. Foi utilizada uma resistência variável a fim de exemplificar o potencial do PSCAD relacionado a simulações em tempo real.

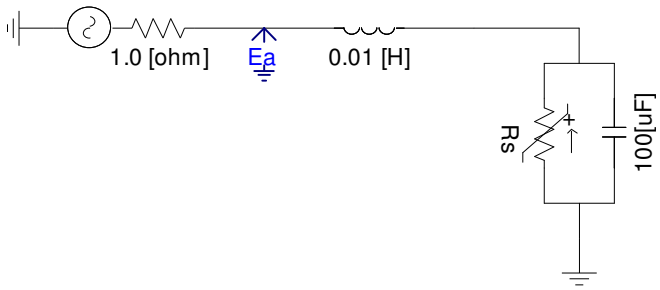



**Figura 4.12 - Exemplo 01 – Circuito para integração**

### 4.4.1. PSCAD

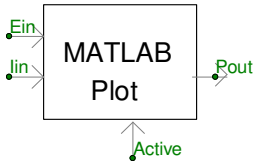



**Figura 4.13 - Exemplo M-File - Componente de integração ao MATLAB no PSCAD**

A configuração dos arquivos deve ser feita de maneira completa e concordante com o arquivo do MATLAB a ser utilizado. No PSCAD, é necessária a criação de um novo componente. Seguindo alguns passos já discutidos na seção 4.3.2, o bloco de integração com o MATLAB foi implementado como visto na Figura 4.13. Foram definidos quatro pontos de conexão para esse componente:

- Ein – Entrada de dados – Valor de leitura de Tensão;

- Iin – Entrada de dados – Valor de leitura de Corrente;
- Active – Entrada de dados – Valor para ativação da integração com MATLAB ou não;
- Pout – Saída de dados – Valor de potência, calculado no MATLAB a partir do produto entre tensão e corrente.

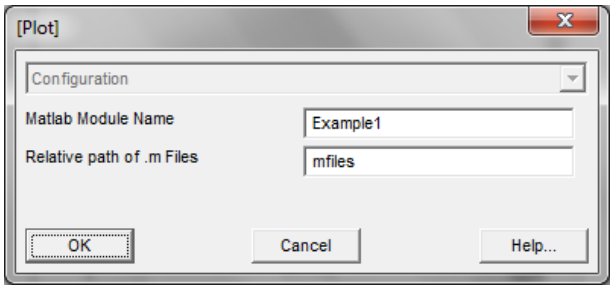



**Figura 4.14 - Exemplo M-File - Parâmetros do componente de integração**

Os parâmetros desse novo componente foram definidas de modo a atender as necessidades básicas para a integração da simulação. Observe na Figura 4.14 os parâmetros definidos. É preciso informar ao PSCAD o endereço relativo ao arquivo do MATLAB, assim como seu nome, a fim de que ele possa encontrá-lo e utilizá-lo.

Na caixa de configuração, a variável "Name" foi definida para o nome do arquivo e a variável "Path" para o caminho relativo do M-File. Esse caminho relativo refere-se ao endereço do arquivo a partir da pasta em que o arquivo de simulação do PSCAD se encontra.

A entrada "Active" do bloco seguirá a lógica de ativar a integração com o MATLAB quando alimentada com um valor unitário e desativar caso seja alimentada com zero. Ela é utilizada no sistema simulado através de uma chave que definirá a utilização ou não da integração, como pode ser visto na Figura 4.15.

Tendo todos estes pontos definidos, deve-se agora definir o que será enviado para o MATLAB. Deve-se sempre ter em mente que a cada passo de simulação no PSCAD, o programa reunirá informações e as enviará para o MATLAB, rodando a M-File e então receberá as respostas. Por isso, a programação deve ser direcionada a este procedimento.

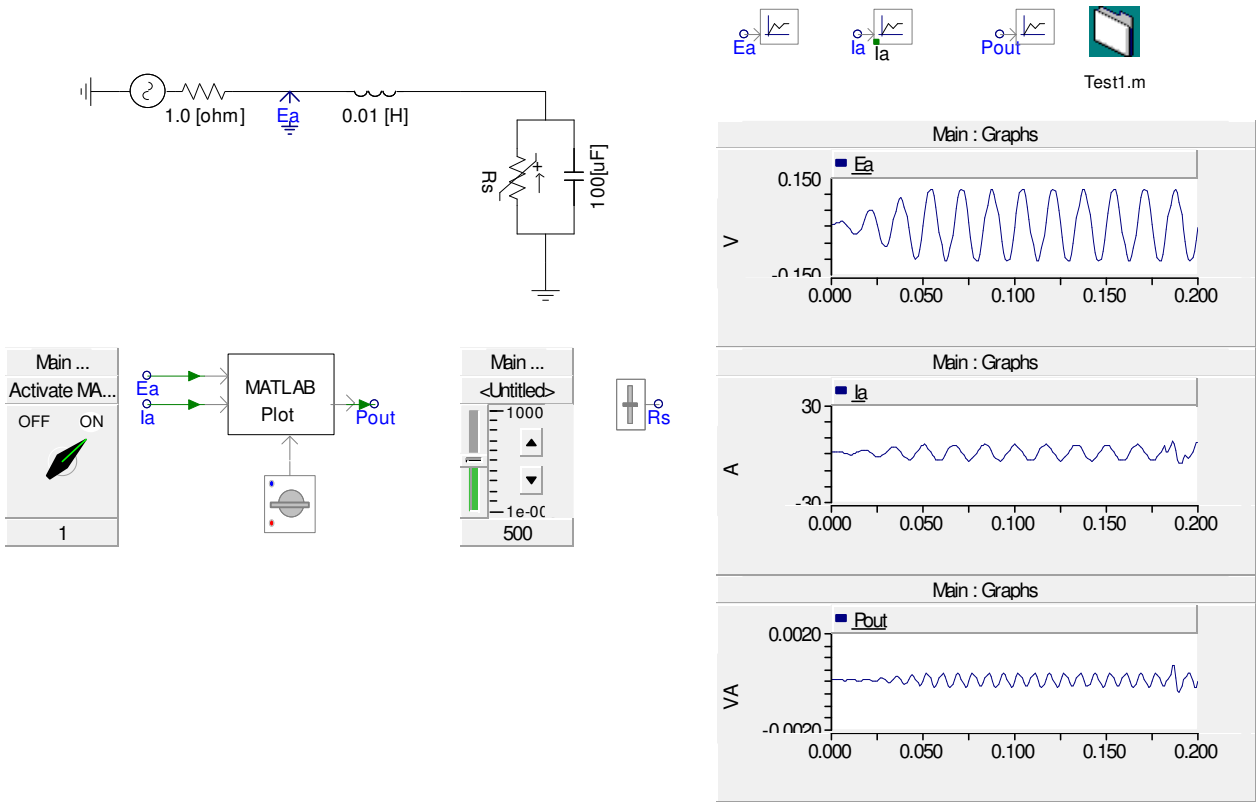



**Figura 4.15 - Exemplo M-File - Sistema completo**

No caso atual são enviados três grupos de dados: o tempo de simulação, não mostrado na figura, uma vez que é enviado através da variável TIME na programação, e a tensão e a corrente recebidas no componente. Esses três grupos são enviados ao MATLAB a cada passo de simulação. O MATLAB então usará o código escrito no M-File e enviará de volta todos os dados definidos.

Todos esses fatores no PSCAD são definidos utilizando a linguagem de programação Fortran. A integração gira em torno da função “MLAB_INT”, reconhecida pelo compilador. A função requer a entrada de quatro parâmetros obrigatórios:

1. O diretório em que se encontra o M-File referente à simulação;
2. O nome dado ao M-File, a fim de se utilizar o arquivo correto;
3. As variáveis a serem enviadas para o MATLAB;

4. As variáveis que retornarão para o PSCAD.

Essa função deve ser chamada da seguinte maneira:

CALL MLAB_INT("MFILEPATH","MFILENAME","Input Format","Output Format")

Dessa maneira, no código exemplo acima, o arquivo com nome “MFILENAME”, presente na pasta “MFILEPATH”, estará sendo utilizado na simulação.

O “Input Format”, ou Formato de Entrada, deve ser definido de maneira a explicitar que tipos de variáveis e suas dimensões que serão enviadas através da integração. Por exemplo, utilizando “R R(31) I” nesse ponto, primeira variável enviada será real de dimensão um, ou escalar, a segunda também real, mas de dimensão 31, e por último uma variável inteira escalar.

O “Output Format”, ou Formato de Saída, é definido de maneira semelhante ao Formato de Entrada. Se neste campo for definido “R(10), é definido que o PSCAD irá receber do MATLAB uma variável de valores reais de dimensão 10.

Para se definir quais as variáveis a serem enviadas, devem-se utilizar os ponteiros “STORF” para valores reais e “STORI” para valores inteiros.

A sua utilização é simples. Por ser um ponteiro, existe um endereço memória que ele aponta. O que se deve fazer é enviar um valor de uma variável para endereços de memória consecutivos, de maneira que ao utilizar a função MLAB_INT, acesse esses endereços e envie seu conteúdo para o MATLAB.

Para enviar três variáveis reais, VAR1, VAR2 e VAR3, e uma inteira, VARI, por exemplo, para uma posição de memória, faz-se uso da posição atual do ponteiro real, definida por NSTORF, e da posição atual do ponteiro inteiro, NSTORI, incrementando gradativamente seus valores, como visto a seguir.

STORF(NSTORF) = VAR1

STORF(NSTORF + 1) = VAR2

STORF(NSTORF + 3) = VAR3

STORI(NSTORI) = VARI

CALL MLAB_INT(“C:\TEMP\MLAB_FILES”, “TEST”, “R R(2) I”, “ ”)

Dessa maneira, serão enviados 3 variáveis para o MATLAB: uma real de dimensão 1 e outra real de dimensão 2 e uma inteira de dimensão 1. Como o último parâmetro da função foi enviado em vazio, o compilador irá interpretar que não há dados de resposta do MATLAB.

Nesse ponto é comum usufruir de laços em código de programação para reduzir o número de linhas do programa, simplificando-o.

Observe que os ponteiros real e inteiro são independentes. A variação do ponteiro real não influencia na posição do ponteiro inteiro.

O mesmo procedimento é feito para o recebimento dos dados enviados do MATLAB para o PSCAD. Os dados estarão armazenados nos endereços posteriores aqueles utilizados para enviar dados. Com isso em mente, deve-se guardar na variável de saída os valores existentes na memória utilizando as funções STORF e STORI, lembrando de utilizar seu valores NSTORF e NSTORI atuais, de maneira semelhante àquela feita para enviar dados.

Ao final do código do programa, é recomendado atualizar os valores dos ponteiros, simplesmente definindo os valores de NSTORF e NSTORI para o valor imediatamente posterior àquele utilizado pela última vez. A partir dessa recomendação do manual do usuário, supõe-se que não haverá problemas em atingir a última posição do ponteiro.

Observe no APÊNDICE B o código desenvolvido em Fortran para o PSCAD para utilização no exemplo 1.

### 4.4.2. M-FILE

No lado do MATLAB, foi desenvolvido em um M-File uma função, para que os valores de entradas sejam reconhecidos, assim como os de saída. Para criar uma função basta criar um arquivo no MATLAB que inicie com o comando:

*Function* [OUT] = Funcao_exemplo([IN])

E salvá-lo com o nome dado à função. As variáveis IN e OUT neste exemplo representam as variáveis de entrada e saída de dados no M-FILE. O M-File receberá as variáveis IN e devolverá as variáveis OUT como responsta.

Dessa maneira, quando o PSCAD enviar dados para o MATLAB, estes serão recebidos pela função, tratados por esta função e então os valores de saída serão enviados de volta ao PSCAD. Vale lembrar que a cada passo dado no PSCAD, o M-File será compilado, simulando aquilo que fora desenvolvido em sua programação.

O código desenvolvido para o exemplo utilizado se encontra no APÊNDICE C.

Observe que no código, foi implementada uma lógica de maneira a não plotar os gráficos em todo passo dado pelo PSCAD, baseando-se no tempo de execução. A razão é que ao fazer o MATLAB plotar gráficos a cada passo de 50 micro segundos diminui muito a velocidade da simulação. Ao limitar o número de vezes que é preciso atualizar os gráficos plotados, a simulação ocorre com maior velocidade.

### 4.4.3. Exemplo 1 – Resultados e comentários

A simulação do exemplo foi executada com êxito, verificando resultados satisfatórios, tanto no PSCAD quanto no MATLAB.

No PSCAD, os gráficos gerados foram obtidos corretamente, incluindo aquele que utiliza dados vindos do MATLAB.

Já no MATLAB, observou-se o sucesso da comunicação entre os dois software. Os dados obtidos foram plotados efetivamente, obtendo-se formas de onda satisfatórias. O envio de dados para o PSCAD, sendo estes o valor de potência calculado no MATLAB através da multiplicação dos valores de tensão e corrente, também se mostrou satisfatória.

Pôde-se observar que o tempo de simulação cresce consideravelmente devido à simulação simultânea em utilizando dois softwares robustos de simulação. Foi possível obter resultados satisfatórios na redução desse tempo de simulação modificando a lógica de programação em um dos softwares, mas isso não é sempre possível.

A simulação completa, junto ao circuito utilizado pode ser visto na Figura 4.16. A Figura 4.17 mostra o resultado obtido no MATLAB, sendo o gráfico em azul a tensão e o em vermelho, dez vezes o valor da corrente.

Vale lembrar que o valor da resistência no circuito foi reduzida de 1000Ω para 0Ω pouco após 0.1 segundos da simulação, o que explica o crescimento nos gráficos da corrente e da potência obtidos.

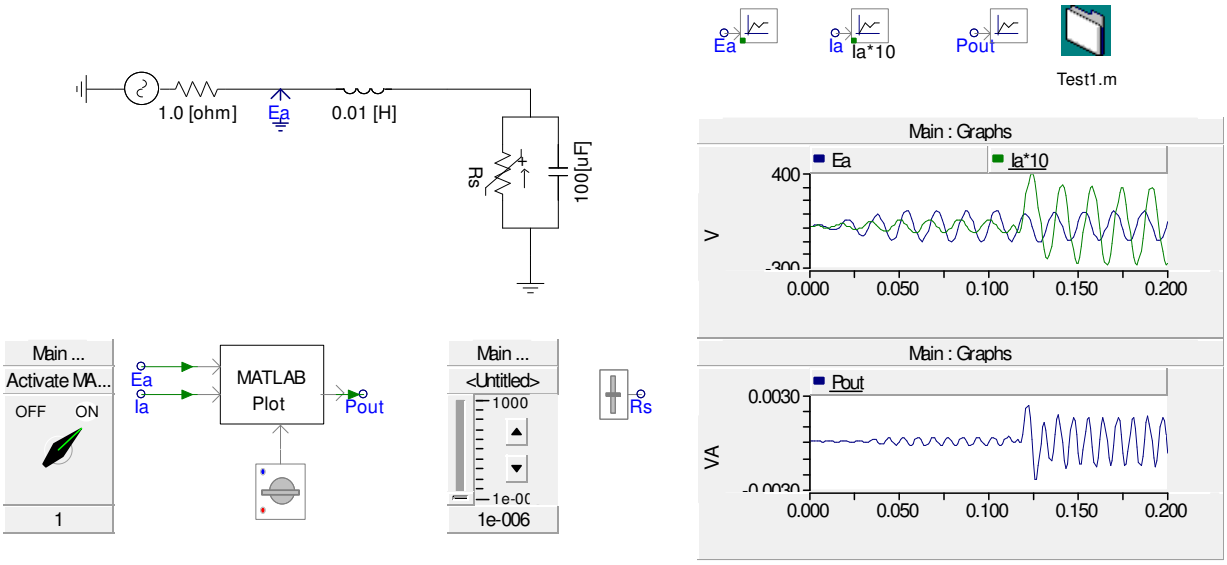



**Figura 4.16 - Exemplo 1 - PSCAD - Simulação completa**

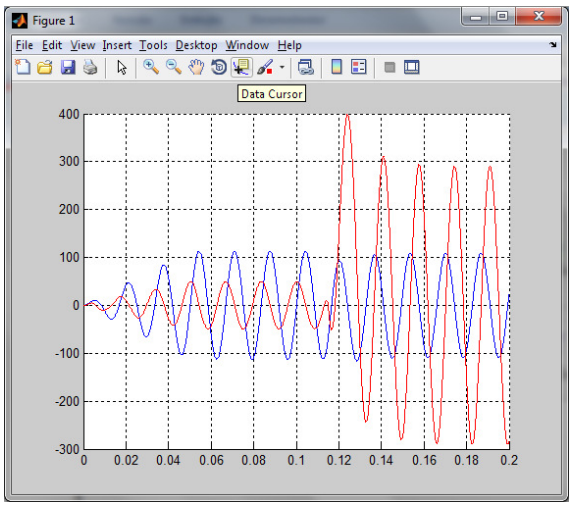



**Figura 4.17 - Exemplo 1 - MATLAB – Gráficos obtidos na simulação completa**

## 4.5. Integração PSCAD – SIMULINK

A integração do PSCAD com o SIMULINK se dá de maneira semelhante àquela com arquivos tipo M-File.

Utilizando o mesmo circuito utilizado no exemplo de integração com M-Files, outro exemplo foi gerado. Dessa vez, o único objetivo fora obter o valor da multiplicação dos valores de tensão e corrente obtidos do PSCAD. Mais uma vez deseja-se mostrar um exemplo meramente explicativo e relativamente simples, o que melhora o entendimento.

### 4.5.1. PSCAD

A função em Fortran a ser utilizada no PSCAD, indicada para integração com o SIMULINK é:

SIMULINK_INT("MFILEPATH","MFILENAME","Input Format").

Observe que nesta função não é definido o Formato de Saída, diferentemente da função MLAB_INT. Esse valor será incrementado à medida que se adicionarem portas de saída de dados na simulação do SIMULINK. É preciso conhecer qual esse valor para recuperar todos os dados retornados, utilizando os ponteiros STORF e STORI, como no caso do M-File.

É preciso atentar, porém, que a primeira variável a ser enviada para o SIMULINK deve ser a variável equivalente ao tempo de simulação. Isso pode ser facilmente efetuado a partir do seguinte segmento de código em Fortran:

STORF(NSTORF + 1) = TIME

Tendo feito isso, o próximo valor enviado utilizando a posição subseqüente do NSTORF será utilizado no primeiro bloco de entrada de dados do SIMULINK.

O código desenvolvido em Fortran para compilar o exemplo em questão, o Exemplo 2, segue anexado no APÊNDICE D.

### 4.5.2. SIMULINK

Para utilizar os dados obtidos do PSCAD no SIMULINK é preciso utilizar um bloco de entrada de dados presente no SIMULINK, nomeado simplesmente de "IN". São semelhantes aos observados na Figura 4.18.

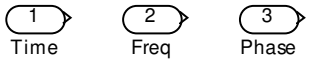



**Figura 4.18 - SIMULINK - Bloco de leitura de dados**

Como se pode imaginar, bloco nomeado "Out" é utilizado para enviar dados de volta ao PSCAD, completando a comunicação de dados.

Ainda é possível utilizar-se da integração do PSCAD com o SIMULINK através da utilização da função MLAB_INT, que chamaria um M-File, e este usaria a função "sim" para lidar com a simulação no SIMULINK. O manual de instruções (MANITOBA, 2005) deixa claro que esse método pode ser usado, mas é altamente aconselhável a utilização da função SIMULINK_INT, devido a mecanismos de sincronização entre os softwares. Isso é especialmente verdade para módulos do SIMULINK cujo tempo de simulação é maior que o passo de simulação do PSCAD.

Ao utilizar o SIMULINK como parte da integração, é normal definir o tempo de simulação nesse software para um pequeno intervalo de tempo, a fim de que os dados sejam interpretados o mais rápido possível e enviados de volta para o PSCAD. Para que os dados não sejam utilizados em período transitório no SIMULINK, também é comum fazer com que a simulação comece em um tempo elevado. Esse parâmetro pode ser modificado na configuração de parâmetros do menu de simulação. Um exemplo desses parâmetros pode ser visto na Figura 4.20.

Observe na Figura 4.19 o circuito desenvolvido no SIMULINK para o Exemplo 2.

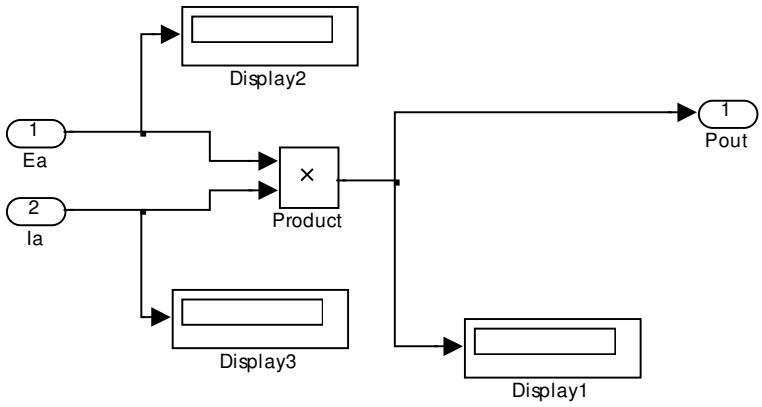



**Figura 4.19 - Exemplo 2 - Circuito SIMULINK**

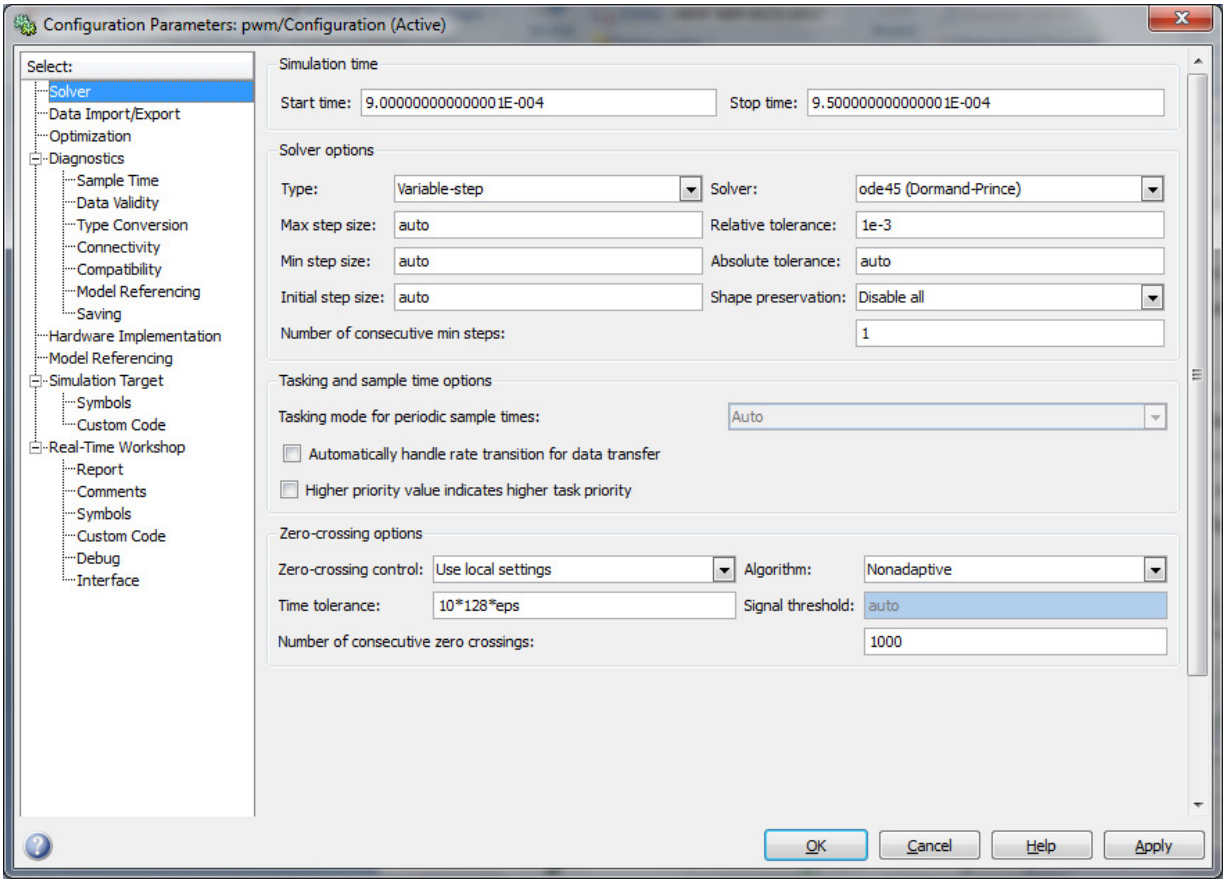



**Figura 4.20 - SIMULINK - Configurações de Parâmetros - Tempo de simulação**

### 4.5.3. Exemplo 2 – Resultados e comentários

Pode ser observado na Figura 4.21 os resultados obtidos a partir da simulação do Exemplo 2, onde se obteve como resposta o valor de potência aparente instantânea.

Observou-se que, mesmo para sistemas simples, montados tanto no PSCAD quanto no SIMULINK, ao integrar suas simulações, o tempo total de simulação foi muito grande. Utilizando um passo de 50 microssegundos e um tempo de simulação de 200 milisegundos no PSCAD e um tempo de simulação de 0,5 microssegundos para o SIMULINK, em um computador de processador Intel Core I3, com 4GB de RAM, a simulação durou cerca de 30 minutos. Isso mostra uma grande desvantagem da integração utilizando SIMULINK, uma vez

que, a simulação feita através do M-File, no Exemplo 1, que exerce as mesmas funções, durou poucos segundos.

Ainda se deve observar que a função exercida pelo SIMULINK neste exemplo pode ser facilmente substituída de diversas maneiras utilizando artifícios do próprio PSCAD, o que mostra que se deve atentar para o que pode ser simulado através desses artifícios, já que se pode obter simulações mais rápidas.

Algumas alternativas são apresentadas para acelerar a simulação. Uma delas é um estudo mais aprofundado da lógica da programação das simulações de maneira a reduzir o número de vezes em que a interface de comunicação entre os dois softwares é acionada. Esse método irá variar para cada simulação, não sendo possível estabelecer um padrão. Um exemplo de como implementar essa alternativa seria através do controle de uma porta de ativação no componente de interface no PSCAD através de um trem de pulsos, como representado na Figura 4.22.

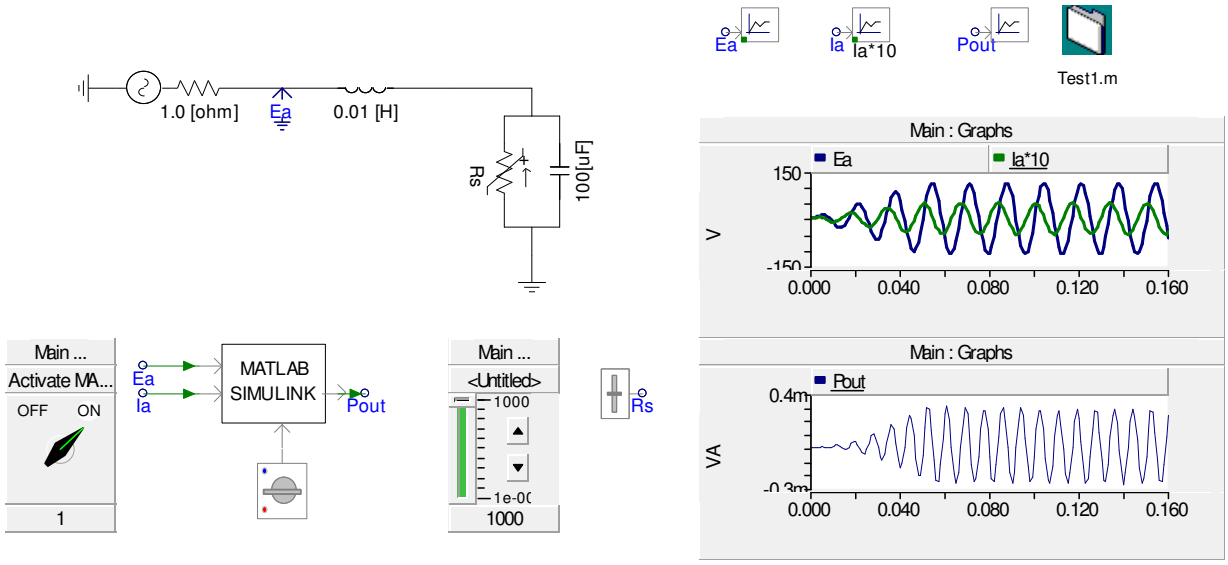



**Figura 4.21 - Exemplo 2 - Resultados**

Outra alternativa, aplicável somente a M-Files, seria a conversão do programa salvo em um arquivo ".m" para a linguagem "C" e então compilar e 'linkar' diretamente o arquivo fonte no executável EMTDC.O Manual do Usuário do PSCAD (MANITOBA, 2005) diz ser possível, embora não tenha sido completamente testado.

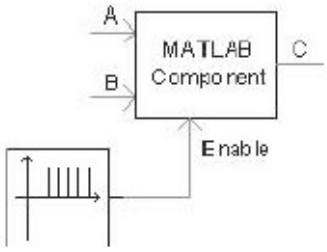



**Figura 4.22 - Exemplo de aumento de velocidade de simulação**

## 4.6. Exemplo PSCAD Support, desenvolvido pelo suporte técnico

As informações obtidas para que se obtivesse sucesso na integração dos dois softwares, especialmente relacionadas a utilização das ferramentas no PSCAD, não foram obtidas somente do Manual do Usuário do PSCAD. O suporte técnico do PSCAD forneceu algumas informações, assim como um exemplo de integração que foi de grande auxílio para a melhor compreensão do método de integração com o SIMULINK.

Esse exemplo consiste em um conversor, cujo controle de ativação de seus tiristores é feito através do controle PWM desenvolvido no SIMULINK. Ele será brevemente apresentado por ser um bom exemplo, podendo ser de grande ajuda no desenvolvimento de outras simulações. É disponibilizada também uma biblioteca que consiste em dois componentes de comunicação com o MATLAB, o que também pode ser útil para futuras aplicações. Na Figura 4.23 tem-se o circuito presente no PSCAD.

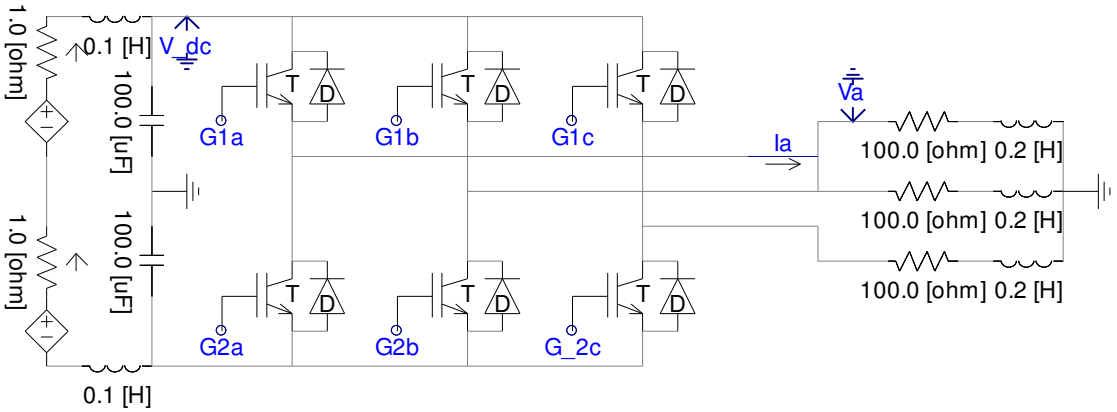



**Figura 4.23 - Exemplo Support - Circuito**

O ANEXO B possui o código em Fortran utilizado no PSCAD.

A Figura 4.24 mostra os controles de simulação dados no PSCAD, além do componente de interface com o MATLAB.

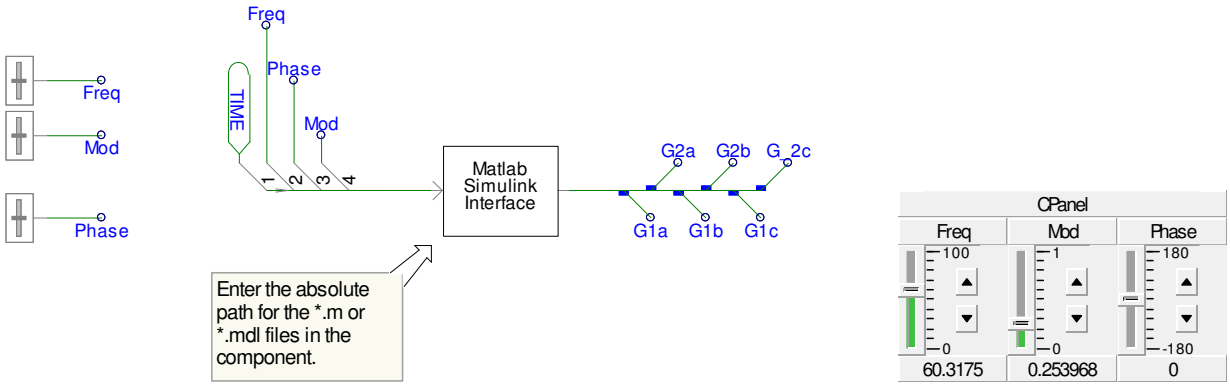



**Figura 4.24 - Exemplo Support - Controles e Componente de interface com o SIMULINK**

Os parâmetros do componente de integração são mostrados na Figura 4.25. Observe que nesse componente é oferecida uma opção de escolha entre a utilização de um arquivo do SIMULINK ou a utilização de um M-FILE para realizar a comunicação entre os dois softwares. Isso também pode ser observado no código desenvolvido em Fortran, disponível no ANEXO B, o qual verifica que opção foi selecionada e através de comandos condicionais “If” (se) decide qual opção é a escolhida.

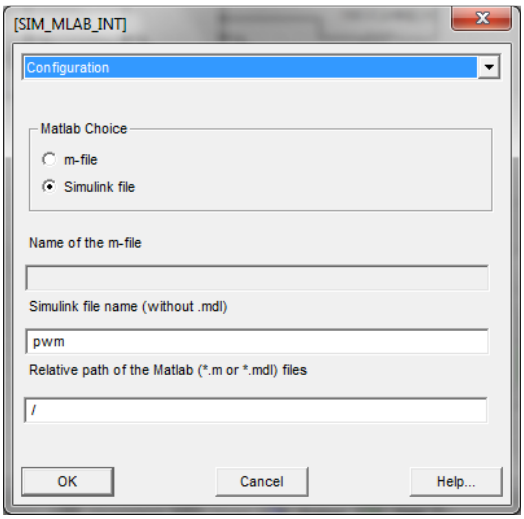



**Figura 4.25 - Exemplo Support - Parâmetros do componente de integração**

Pode-se observar também que o bloco utilizado para a interface entre os softwares é sobremaneira genérico, podendo ser utilizado em qualquer simulação. Isso ocorre por serem utilizados vetores como parâmetros de entrada e saída de dados, o que é possível graças à utilização de laços limitados pelo valor da dimensão dos vetores em questão. É definido uma dimensão igual a zero para o pino de entrada, para que o PSCAD aceite valores variáveis nessa dimensão. No caso do pino de saída, é necessária a definição de sua dimensão. Por isso, para utilizar este componente em outra simulação que requer integração com o SIMULINK, esse valor necessita ser redefinido.

A Figura 4.26 mostra o sistema montado no SIMULINK, cuja finalidade é enviar para o PSCAD o controle PWM.

Ao simular o sistema utilizando o mesmo computador utilizado anteriormente, com um processador Intel Core I3, com 4GB de memória RAM. Foi definido um passo de 50 microssegundos, durante um intervalo de 0,5 segundos. O tempo de simulação utilizado no SIMULINK variou de 0,9 milissegundos até 0,95 milissegundos, tendo assim uma duração de 50 microssegundos. O tempo total de duração de simulação foi de aproximadamente 3 horas, o que mostra mais uma vez a grande desvantagem desse método de integração.

Para que seja verificado o sucesso da simulação, segue na Figura 4.27 o sistema com a simulação concluída. Também se observa os gráficos gerados no PSCAD, mostrando os valores de tensão e corrente do lado CC, e os valores de tensão de fase e corrente do lado da carga.

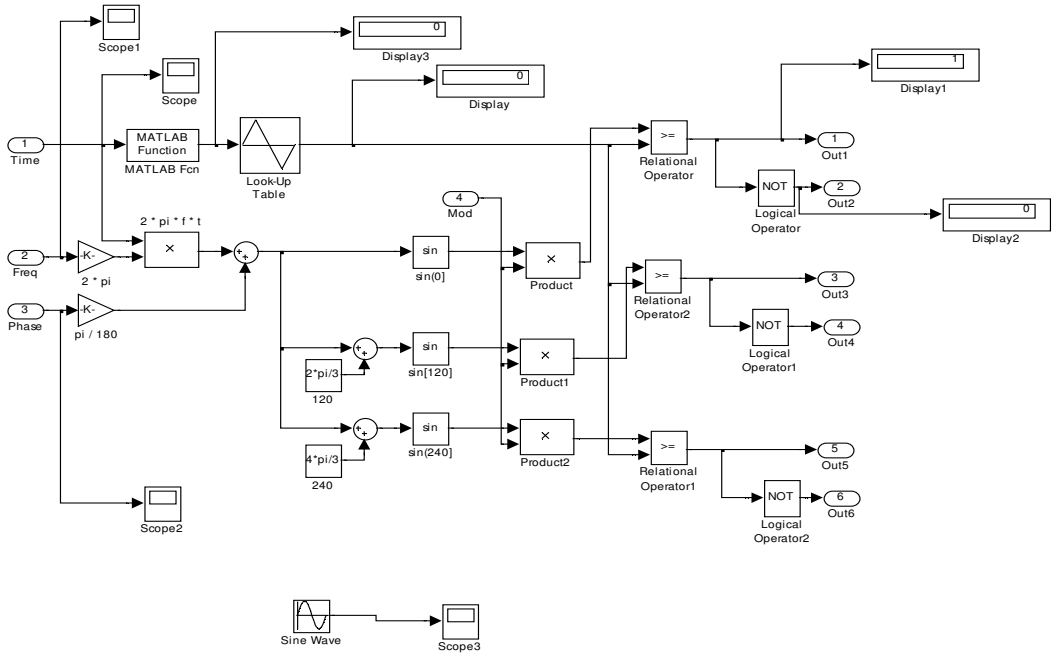



**Figura 4.26 - Exemplo Support - Sistema no SIMULINK**

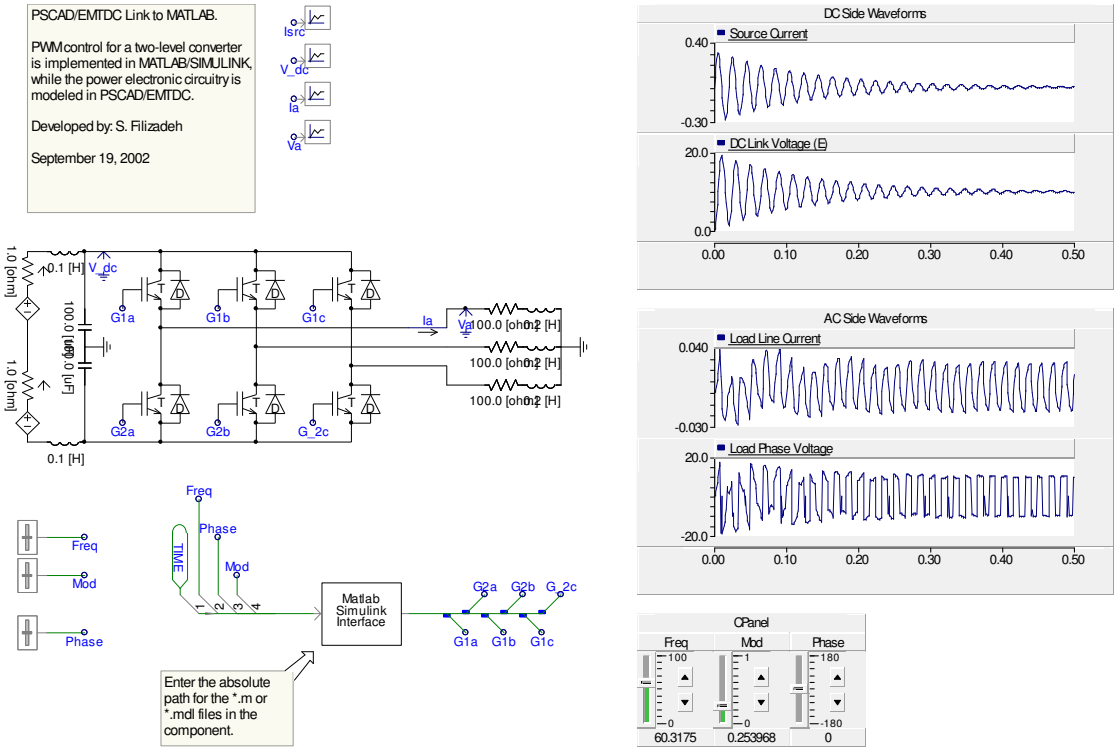



**Figura 4.27 - Exemplo Support - Sistema simulado - PSCAD**

## 4.7. Considerações Finais

A integração entre PSCAD e MATLAB pode ser muito proveitosa, visto que um software possui diversas funções que o outro não possui. Porém deve-se estar ciente de suas limitações antes de utilizá-la, uma vez que a limitação de velocidade de simulação pode ser um grande problema em alguns casos.

Os aprimoramentos nessa velocidade podem reduzir os problemas encontrados na integração, principalmente com o SIMULINK. Para isso, deve-se conhecer mais profundamente as ferramentas disponibilizadas pelo PSCAD, a fim de amenizar a quantidade de dados a ser tratada no SIMULINK, tratando-a localmente no PSCAD.

A existência de exemplos da integração disponibilizados na instalação do PSCAD foi de grande importância para o conhecimento obtido no decorrer do projeto. Houve também algumas dificuldades na compreensão da integração com o SIMULINK até que o exemplo enviado pelo suporte técnico foi obtido. Isso se deu principalmente pela escassez de informações sobre o assunto. Indubitavelmente, o auxílio do suporte técnico foi de indispensável ajuda para o desenvolvimento deste trabalho.

# Capítulo 5

# Conclusões

A utilização de simulações é indispensável na engenharia. São vários os softwares existentes no mercado que visam facilitar a compreensão, análise e melhoria de sistemas através de simulação computacional. Cada software tem suas peculiaridades e se diferenciam pelas funções que oferecem ao usuário, pelo conteúdo de suas bibliotecas, pelo tamanho do passo de simulação, pelo tamanho do sistema que são capazes de representar, etc. Poder aproveitar códigos de programas entre softwares de engenharia tem como vantagem o aproveitamento de horas-de-trabalho e o uso compartilhado de benefícios que cada software dispõe individualmente.

Em sendo os softwares PSCAD e MTLAB/SIMULINK muito usados na engenharia elétrica, e em particular na área de sistemas de potência, buscou-se neste trabalho uma maneira de aprimorar essas simulações de modo a utilizar alguma comunicação entre os dois softwares.

A comunicação entre os softwares pode trazer grandes vantagens, uma vez que cada software possui características próprias que os distinguem entre si. No caso do PSCAD, é possível realizar simulações complexas de grandes sistemas de potência, além da capacidade de modificação de parâmetros no decorrer da simulação. Pelo MATLAB é possível simular componentes com maior detalhe, com passo de simulação bem menor que no PSCAD, não sendo, no entanto possível representar sistemas elétricos com muitas barras. No MATLAB há ainda a capacidade de manipulação de dados, simplicidade de componentes, a possibilidade de modificá-los, além da capacidade de criar gráficos de maior complexidade, como por exemplo, gráficos em três dimensões.

Embora vários problemas e obstáculos tenham sido encontrados durante a instalação e integração dos softwares, se obteve êxito em integrar completamente as simulações. Com isso, foi possível criar um passo a passo a fim de facilitar futuras instalações do software junto ao compilador, além de criar também uma pequena relação de erros a ser corrigidos. Isso poderá auxiliar futuros usuários do sistema, que poderão utilizar esse material como ponto de partida para pesquisas futuras, reduzindo assim a escassez de material relacionado a este assunto.

Ao utilizar a comunicação entre os softwares, entretanto, observou-se uma redução considerável da velocidade de simulação. O tempo de simulação para sistemas que utilizam M-Files é maior que uma simulação simples utilizando somente o PSCAD. O tempo de simulação se mostrou muito maior para sistemas que utilizaram o SIMULINK como ferramenta de simulação no MATLAB.

Existem métodos de reduzir este tempo de simulação recomendados pelos desenvolvedores, como por exemplo, limitar no PSCAD o número de vezes em que é efetuada a comunicação com o MATLAB, através de configurações de ativação da interface, ou a tradução do código em MATLAB para C, adicionando-o ao arquivo do PSCAD, o que levaria à redução considerável do tempo de simulação, uma vez que o código originalmente desenvolvido no MATLAB agora faria parte do arquivo do PSCAD e seria simulado nesse software.

Fica a sugestão para que estudos mais aprofundados sejam desenvolvidos relacionando maneiras mais rápidas de utilizar essa interface PSCAD – MATLAB.

Tendo conhecimento do acréscimo de tempo de simulação, principalmente utilizando o SIMULINK, deve-se então ter a certeza da real necessidade dessa integração, no caso de o tempo de simulação representar um problema. Algumas vezes, as ferramentas disponibilizadas pelo PSCAD podem ser suficientes para substituir algumas funções do MATLAB. Por isso, o conhecimento mais aprofundado dessas ferramentas pode ser de fundamental importância em algumas simulações.

# Referências Bibliográficas

DUARTE; AVELAR. Avelar e Duarte, 2010. Disponivel em: <http://www.avellareduarte.com.br/projeto/conceitos/projeto/projeto.htm>. Acesso em: 2 Maio 2011.

FARHAD SHAHNIA, M. B. B. S. **Pscad/Emtdc Based Simulation Of Double Fed Induction Generator For Wind Turbines**. University of Tabriz. Tabriz, Irã.

FRAMEWORK, I. N..NET FrameWork: Overwiew.**.Net Framework**, 2011. Disponivel em: <http://www.microsoft.com/net/overview.aspx>. Acesso em: 12 Maio 2011.

INTEL®. Intel® Software Network. **Intel Visual Fortran - fatal error LNK2023:** bad DLL or entry point msobj80.dll, 2009. Disponivel em: <http://software.intel.com/en-us/articles/ivf-fatal-error-lnk2023-bad-ddl-or-entry-point-msobj80dll/>. Acesso em: 23 Abril 2011.

JIAYI, H.; CHUANWEN, J.; RONG, X. **A review on distributed energy resource and MicroGrid.** SHANGHAI JIAOTONG UNIVERSITY. Shangai, China.

MANITOBA, H. R. C. **PSCAD User's Guide**. Winnipeg, Manitoba, Canada: [s.n.], 2005.

MÁQUINASVIRTUAIS. O que são Máquinas Virtuais? **TecMundo**, 2008. Disponivel em: <http://www.tecmundo.com.br/232-o-que-sao-maquinas-virtuais-.htm>. Acesso em: 12 Maio 2001.

TECHNET, M. Matlab in Windows 7. **TechNet**, 2009. Disponivel em: <http://social.technet.microsoft.com/Forums/en-US/w7itproappcompat/thread/4dba5d57-3127-48f3-9461-eb1ef5d7c70e/>. Acesso em: 12 Maio 2011.

USKI SANNA, L. B. K. J. R. S. A. P. **Adjoint wind turbine modeling with ADAMS, Simulink and PSCAD/EMTDC**. CHALMERS UNIVERSITY OF TECHNOLOGY. Finlandia. 2004.

VISUALSTUDIO, I. -. Visual Studio Home. **Microsoft Visual Studio**, 2011. Disponivel em: <http://www.microsoft.com/visualstudio/en-us/home>. Acesso em: 12 Maio 2011.

# APÊNDICE A

## Auxílio do Suporte Técnico do Software PSCAD via e-mails

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: February-23-2011 10:55 AM
Subject: Intel Visual Fortran Compiler

Hello,

We purchased the PSCAD v4.2.1 and we need to compile MatLAB simulations into our simulations. For that we purchased the Intel® Visual Fortran Compiler, but after installing it, no new option appeared under the Fortran TAB under *Workspace Settings*. How should I proceed to make the link between the two softwares? Can you help us with this asap?

Thanks,

Guilherme Nascimento
Universidade Federal do Ceará

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: February-23-11

Hello Guilherme

You can download the "FortranMedic" from our site.

http://pscad.com/updater/utilities/FortranMedic.zip

The following are instructions on how to use the FortranMedic utility:
1) Download the ZIP from the FTP site
2) Run the FortranMedic.exe file
3) Click on the |Actions|menu and select "Start"
4) After the utility is done retrieving information, you have 2 options

**Option 1**
You can right click on any text in red and select OK from the dialog to repair the configuration errors

**Option 2**
Click on the |File|menu and select "Save Messages"

The FortranMedic message window should now display the location of the 'messages' text file on your computer.

Please send me this file so I can determine what may be wrong with your setup.

Please let me know if you were able to resolve this.

Regards

---

George Wai
PSCAD Software Development and Support
Manitoba HVDC Research Centre
A division of Manitoba Hydro International Ltd.
211 Commerce Drive
Winnipeg, Manitoba
Canada, R3P 1A3
Tel. +1 204-989-1240
Fax. +1 204-989-1277
www.hvdc.ca

Kindest Regards

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: March-02-2011 09:30 AM
Subject: Intel Visual Fortran Compiler

Thanks for that. I tried to use it and fix all the problems, but when I did it, I couldn´t run any simulation. I am sending you the file generated by the FortranMedic. I think that there is something wrong with Visual Studio 2008. Can you help me solving this?

And is there any document that teaches how to create a link between those MatLab and PSCAD? I tried somethings but I don´t know if it would work.

Also, I have the PSCAD 4.2.1 and we received an email from CEDRAT saying to upgrade it to the 4.3.1 in the PSCAD website (www.pscad.com). I couldn´t find the link to download it. Can you help me with this also?

Some info about the purchase, if needed:
PO #10,025
In the name of: Astef - UFC

If you need anything, please ask.

Thanks for your help,
Guilherme

---

From: PSCAD Support Team <support@pscad.com>

To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-02-11

Hello Guilherme
You need to uninstall Compaq Fortran because it is interfering with the operation of the Intel compiler.
Please let me know if this works for you
Regards

_____
George Wai

---

From: Frankie S. Wasylik fsa@hvdc.ca
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-03-11
Subject: RE: (Case 11320):Intel Visual Fortran Compiler setup problem

Dear Guilherme,

I have registered you on our website at www.pscad.com so that you can download the PSCAD X4 software. Please log on to your account using the following:

User ID: *********
Password: *********

Please let me know if you have any questions.

Thanks and have a great day!
Frankie

**Frankie Wasylik**,
Customer Service Team Leader
**Manitoba HVDC Research Centre**,
a division of Manitoba Hydro International Ltd.
211 Commerce Drive
Winnipeg, Manitoba
Canada, R3P 1A3
T: +1 204 989 1243
F: +1 204 989 1277
E: fsw@hvdc.ca

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: March-02-2011 09:30 AM
Subject: Intel Visual Fortran Compiler

I tried uninstalling it but it didn´t work... this is the new medic file

Guilherme

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-03-11

Dear Guilherme Martins,

Can you please send us a snapshot of the related error messages that you see?

**Situation**

- I see that PSCAD 4.2.1 is installed with GNU Fortran and with Intel Fortran 11.1.067 and the Premier Partner Edition of Visual Studio (comes with Intel 11.1)

- I also see remnants of the Compac installation, folders left behind after Compac was uninstalled.

  We know these folders exist, otherwise the Medic would have flagged them as orphan folders.
  I don't know what the folders contain, but suspect that their contents are interfering with the Intel compiler

- I also see that the 64-bit portion of the Intel compiler has been installed

**Solution**

You have 3 choices here:

1. Manually delete these folders, but that could impact other programs

2. Manually move the Compac-related folders to the end of the PATH and LIB environment strings, so that their priority and influence are greatly diminished

3. Use the latest FortranMedic to move the Compac-related folders to the end of the PATH and LIB environment strings

I recommend option 3 above.

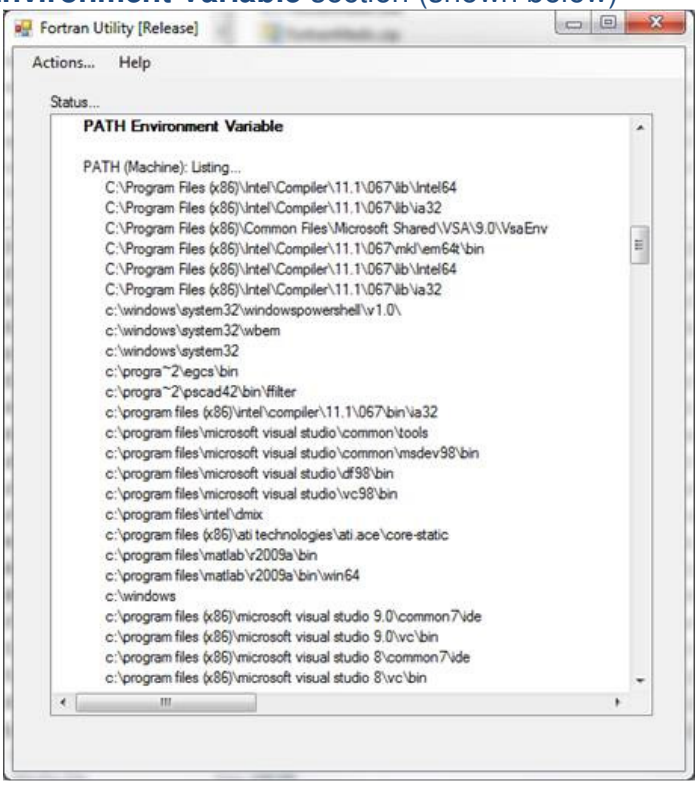
1. Make sure PSCAD is not running

2. Again, download the latest FortranMedic (yesterday's version does not have this capability) from

http://pscad.com/updater/utilities/FortranMedic.zip

3. Run the downloaded Medic again

4. In the **PATH Environment Variable** section (shown below)




5. Right click on the following segments and let the Medic move them to the end of the PATH environment variable value

c:\program files\microsoft visual studio\common\tools
c:\program files\microsoft visual studio\common\msdev98\bin
c:\program files\microsoft visual studio\df98\bin
c:\program files\microsoft visual studio\vc98\bin

6. **In the LIB Environment** section

Right click on the following segments and let the Medic move them to the end of the LIB environment variable value

C:\Program Files\Microsoft Visual Studio\DF98\IMSL\LIB
C:\Program Files\Microsoft Visual Studio\DF98\LIB
C:\Program Files\Microsoft Visual Studio\VC98\LIB

7. Run the Medic again to verify that the segments above are at the end of the list

8. Try to run PSCAD again

9. If it doesn't work, exit PSCAD, use the Medic to move the following Intel Fortran (64-bit- related) segments to the end

In the **PATH Environment Variable** section, right click on the following segments and let the Medic move them to the end of the PATH environment variable value

C:\Program Files (x86)\Intel\Compiler\11.1\067\lib\Intel64
C:\Program Files (x86)\Intel\Compiler\11.1\067\mkl\em64t\bin
C:\Program Files (x86)\Intel\Compiler\11.1\067\lib\Intel64

10. Try to run PSCAD again. If that does work please send us the latest Medic messages file.

Regards,

Lawrence Arendt, P. Eng.
PSCAD Product Development & Support

**Manitoba HVDC Research Centre**, a division of Manitoba Hydro International Ltd.

211 Commerce Drive
Winnipeg, Manitoba
Canada, R3P 1A3

Fax: +1 204 989 1277
Website: www.pscad.com; www.hvdc.ca

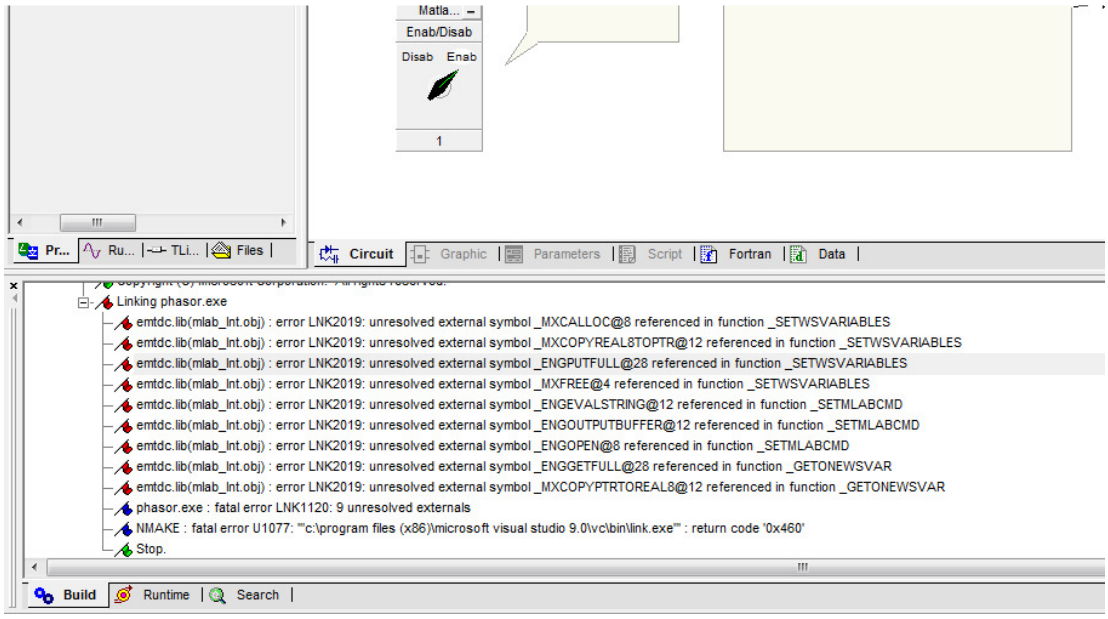
From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: March-04-11 12:38 PM

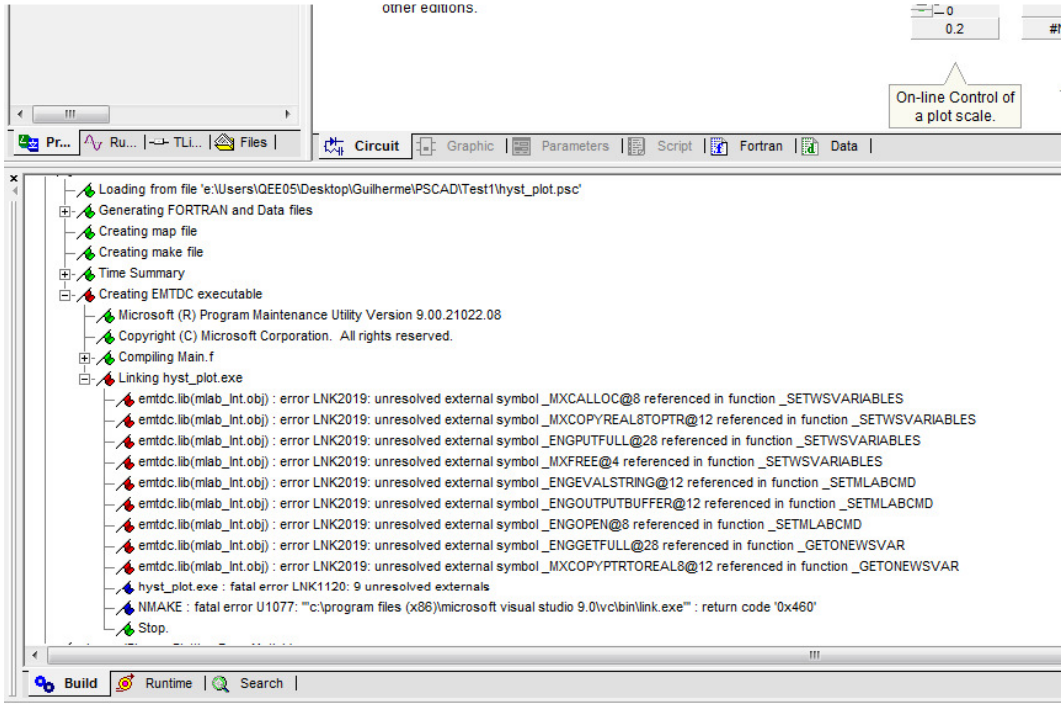
I did until step 8 and when I tried, it worked fine. Now I am able to run a simple simulation, but when I tried to run some of the matlab examples that come with the PSCAD, just to be sure that it is working fine, but I couldn´t... there is something wrong with the link.exe... I am sending you some printscreens of the error. I´ll also send the report from FortranMedic just in case you need it.

I really apreciate all this help.

Thanks,
Guilherme Martins

Anexos:


Matla...
Enab/Disab
Disab Enab
1
Pr...
Ru...
TLi...
Files
Circuit
Graphic
Parameters
Script
Fortran
Data
Linking phasor.exe
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _MXCALLOC@8 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _MXCOPYREAL8TOPTR@12 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _ENGPUTFULL@28 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _MXFREE@4 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _ENGEVALSTRING@12 referenced in function _SETMLABCMD
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _ENGOUTPUTBUFFER@12 referenced in function _SETMLABCMD
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _ENGOPEN@8 referenced in function _SETMLABCMD
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _ENGGETFULL@28 referenced in function _GETONEWSVAR
emtdc.lib(mlab_lnt.obj) : error LNK2019: unresolved external symbol _MXCOPYPTRTOREAL8@12 referenced in function _GETONEWSVAR
phasor.exe : fatal error LNK1120: 9 unresolved externals
NMAKE : fatal error U1077: ""c:\program files (x86)\microsoft visual studio 9.0\vc\bin\link.exe"" : return code '0x460'
Stop.
Build
Runtime
Search


other editions.
0.2
On-line Control of a plot scale.
Pr... Ru... TLi... Files
Circuit Graphic Parameters Script Fortran Data
Loading from file 'e:\Users\QEE05\Desktop\Guilherme\PSCAD\Test1\hyst_plot.psc'
Generating FORTRAN and Data files
Creating map file
Creating make file
Time Summary
Creating EMTDC executable
Microsoft (R) Program Maintenance Utility Version 9.00.21022.08
Copyright (C) Microsoft Corporation. All rights reserved.
Compiling Main.f
Linking hyst_plot.exe
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _MXCALLOC@8 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _MXCOPYREAL8TOPTR@12 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _ENGPUTFULL@28 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _MXFREE@4 referenced in function _SETWSVARIABLES
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _ENGEVALSTRING@12 referenced in function _SETMLABCMD
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _ENGOUTPUTBUFFER@12 referenced in function _SETMLABCMD
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _ENGOPEN@8 referenced in function _SETMLABCMD
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _ENGGETFULL@28 referenced in function _GETONEWSVAR
emtdc.lib(mlab_Int.obj) : error LNK2019: unresolved external symbol _MXCOPYPTRTOREAL8@12 referenced in function _GETONEWSVAR
hyst_plot.exe : fatal error LNK1120: 9 unresolved externals
NMAKE : fatal error U1077: ""c:\program files (x86)\microsoft visual studio 9.0\vc\bin\link.exe"" : return code '0x460'
Stop.
Build Runtime Search

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-04-11

Dear Guilherme Martins,

Situation

It appears that you have installed the 64-bit version of Matlab R2009a.

Requirements

1. PSCAD 4.2.1 is a 32-bit application and requires the 32-bit installation of Matlab; it is not compatible with the 64-bit version

2. Additionally, Matlab R2008b is the last version that is compatible with PSCAD 4.2.1 . You have Matlab 2009a which is not compatible.

Solutions

1. Uninstall Matlab R2009a if it is not needed

2. Install Matlab 2008b, which should be available from the Matlab website.

3. Setup PSCAD 4.2.1 as per the attached document, extracted from the Help manual

4. Run PSCAD 4.2.1

Alternatively, you could install and use PSCAD X4, which is compatible with Matlab 2006a and better. See the attached compatibility document.

Regards,

Lawrence Arendt, P. Eng.
PSCAD Product Development & Support

**Manitoba HVDC Research Centre**, a division of Manitoba Hydro International Ltd.

211 Commerce Drive
Winnipeg, Manitoba
Canada, R3P 1A3

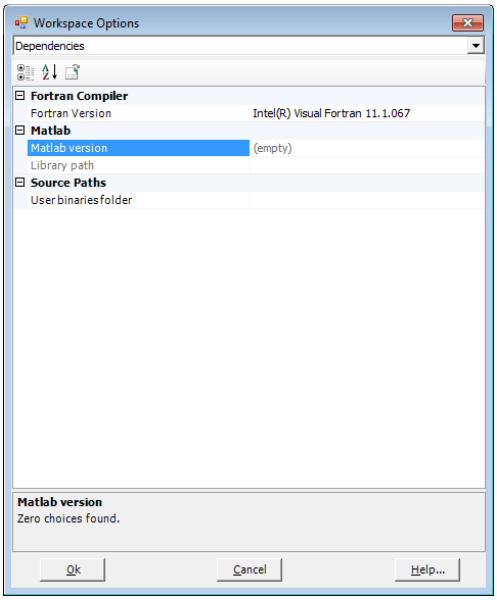
Fax: +1 204 989 1277
Website: www.pscad.com; www.hvdc.ca

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: March-11-11 9:03 AM

Lawrence ,

I tried to install the newer PSCAD X4, and now everything but the link with MATLAB is working. I tried reinstalling the PSCAD but it didn't work. What can be done to solve this?




---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-11-11

Dear Guilherme Martins

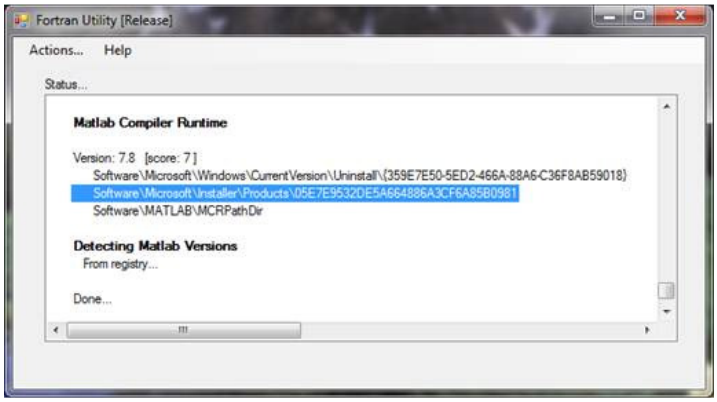
**Item 1**

In order to use MATLAB with PSCAD, you need install a Matlab Compiler.
Since you have Intel 11.1, you can install and use Matlab R2008a (7.6) or better.

PSCAD X4 should automatically detect and list all such versions and better, allowing you to select which Matlab version you want to use.

**Item 2**

From the Fortran Medic log file, you have installed the Matlab Compiler Runtime 7.8 which is only required if you want to use advanced curve fitting for T-Line models




I hope this helps,

Lawrence Arendt

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-15-11 12:18 PM
To: PSCAD Support Team

Dear Lawrence,

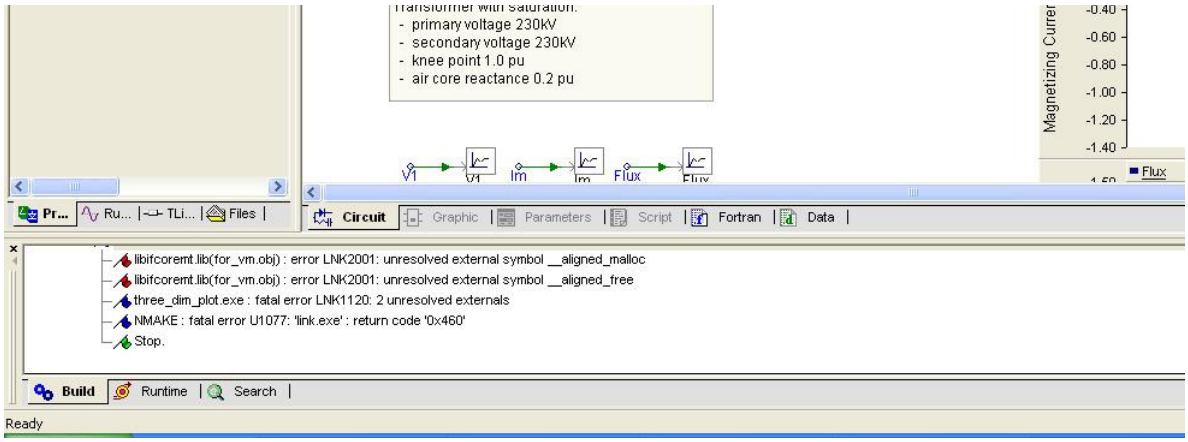
Thank you for all your help with this. I finally decided to change computers and go to one with Windows XP installed.
Once I did this, evereything went way easier. I managed to install and see the Visual Fortran really fast. I only had to install Visual Studio.

Unfortunatly, when I try to run the matlab examples, I receive the errors LNK2001, LNK1120 and LNK1077, as can be seen in the atached image.
I searched for it in the internet and I read that there is something missing in the LIB enviromental workspace, or something like this. So I ran the FortranMedic and there are some red lines, so I am sending you the report so you can help me with this. Hopefully this will be the last problem.

Thank you,

Guilherme

Anexos




Anexo

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-15-11

Dear Guilherme Martins,

I think you missed step 5 as described in the attached document.
Please note that your installation path might be different than the one used in the document's example.

Also please note the updated support case # in the e-mail subject line

Regards,

Lawrence Arendt

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: March-15-11

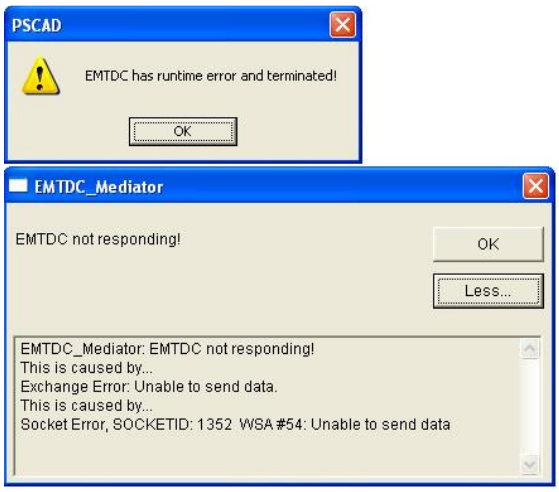
Lawrence Arendt,

Yes, I had missed that. Now it gives me the error in the attached image. And a MATLAB Comand Window opens and stay opened. This only happens when runing examples with MATLAB linked (the examples that come with PSCAD). Other simulations are running correctly.

Thanks

Guilherme Martins

Anexos:




Anexo

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-15-11

Hello Guilherme

I have studied the correspondence between you and Lawrence and I would like to offer my help. Your symptoms and error message indicates that Matlab did not start up and left PSCAD waiting thus an error was thrown. I would like you to try to start Matlab on its own and let me know if this was successful. This will help me narrow down your problem.

I look forward to hearing from you soon

Regards

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-16-11

To: PSCAD Support Team

Hi George,

Thanks for helping with this.
I tried to start MATLAB before running the example in PSCAD and I received the same error.
If it help, beside the error, a "MATLAB Command Window" pops up and stays open, all blank (and it runs the MATLAB commands that I wrote on it).
Is there anything that I can do to help you to help me?

Thanks,

Guilherme Martins

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-17-11

Hello Guilherme

Your PSCAD is configured correctly to use Matlab and from what you are telling me, your Matlab is installed correctly because you can start Matlab on its own and use it independently. Everything should work, the only thing that looks suspicious to me is the fact that you have everything installed in D: drive. This is an untested scenario and I am not sure if this works. It may be the source of your problem because all of our users typically install software to C: drive.

Is your operating system on C: or D: drive?

Regards

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: March-17-11

George,

I have windows XP installed on D: drive

maybe I can try to use a virtual machine or something like that, no?

---

From: PSCAD Support Team <support@pscad.com>

To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-18-11

Hello Guilherme

Please let me know if you were successful installing PSCAD and Matlab on your virtual machine on a C: drive.

Regards

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: March-30-11

Dear George Wai.

I had some problems with the internet here, so I wasn´t able to download somethings.

So,
I managed to install a virtual Machine.
Installed a Win XP on it,
Installed MATLAB 2008b
Downloaded the .net 4 Framework
Downloaded the Microsoft SDK pack
Tried to install the PSCAD with Fortran and it didn´t work,

so I though that I might need the Microsoft Visual Studio
So I downloaded the Microsoft Visual Studio 2010

With everything installed I reinstalled PSCAD 4.2 and the Intel Fortran Visual Compiler.

I´m still not able to simulate anything on PSCAD by using this compiler. I receive an error when linking the file.

I´m also not capable of doing the steps given me on other emails, once the lines in the Fortran Medic don´t match with the ones over there. I am sending you the log generated by the Medic.

Can you help me with this? There has to be something that I am doing wrong, since I installed it on a completely clean win xp on the C drive.

Maybe I didn´t download something vital or something like this....

Otherwise Can you see what is wrong with the one installed in the Virtual Machine?

I am getting really frustrated with all of this.

Thank you,

Guilherme Nascimento

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: March-30-11

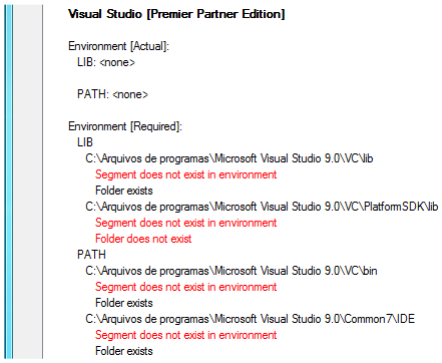
Hello Guilherme

The Intel Fortran 11.1 does not support Visual Studio 2010 according to their website. Also according to your log file, you are missing the Visual studio 2008 SDK that comes with Intel 11.1
I would suggest that you please do the following:

1) Uninstall Visual Studio 2010

2) Reinstall Intel 11.1 but this time, make sure you install all of the Visual Studio 2008 components (specifically the SDK)

Then run the Fortran Medic again and this time right click on the errors in red




If this does not work, please send me a new Fortran Medic log.

I suspect that you are very close to fixing all of your problems. We are here to help you through the entire process

Regards

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: April-05-11

George,
Great!
Now it is working just fine with the examples in PSCAD folder!
I really appreciate all your help!

Now I can start working on it.

Thank you for everything. I´ll try to install the PSCAD 4.3 in the computer with Windows 7, and I will contact you if I need any help, if there isn´t any problems.

Thanks!

Guilherme Nascimento

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: April-24-11

Hi,

I am having trouble on learning how to comunicate PSCAD with SIMULINK, and you told me once that I could look for you for some assistence, so, here I am.
I already did some comunication with m files, but I am a little confused on knowing how to get data from PSCAD on SIMULINK. Is there a specific blok to add? For example, in the m file you get the data as input variables of a function, but how does this work with SIMULINK files?
Can you help me with this?

And one other question, can I send a eletrical signal from MATLAB to SIMULINK, so I can use something already done in SIMULINK to power some loads in PSCAD? I´m not sure if this is possible...

I appreciate your attention,

I am looking forward for an answer

Thank you!

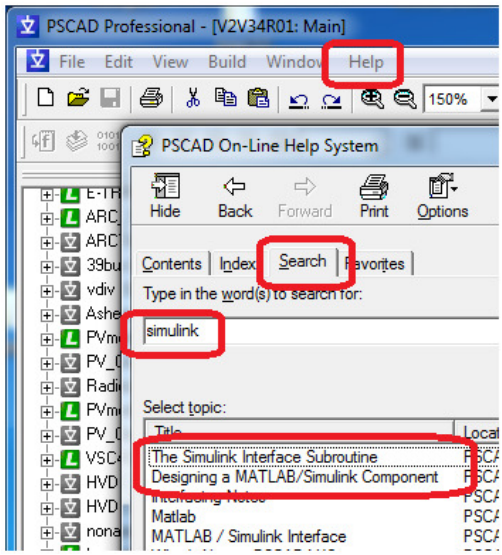
Guilherme Nascimento

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: April-28-11

Hello Guilherme

I have attached a great example of how to interface with Simulink. You also find instructions on how to do it in the PSCAD help.




Regards

---

From: Guilherme Martins [mailto:guimgn@gmail.com]
To: PSCAD Support Team
Sent: May-02-11 11:43 AM

George,
This really helped! Thank you for your support!
Just one more quick questions for you:
Can I send a electrical signal from SIMULINK to PSCAD? Let's say I want to power a load in PSCAD with a power supply built in Simulink?

Sorry, I got it wrong last email.. my bad..

Thanks for your help!

Regards,

---

From: PSCAD Support Team <support@pscad.com>
To: Guilherme Martins [mailto:guimgn@gmail.com]
Sent: May-03-11

Hello Guilherme

You can do this but not directly, you can get the values from Simulink and set the value of voltage/current source and/or RLC values.

Regards

# APÊNDICE B

## Código desenvolvido em Fortran para PSCAD para Exemplo 1

```
#STORAGE REAL:3
! ------------------------------------------------
! ------------------------------------------------
! PSCAD / MATLAB INTERFACE
! Modulo: $Name
! Definindo uma variável inteira "count"
#LOCAL INTEGER count
!
! Verifica se o parâmetro "Enabl" esta igual a 1
      IF($Active.GT.0.9) THEN
!
! ------------------------------------------------
! Enviando os parâmetros das variáveis para o MATLAB
! ------------------------------------------------
! Entrando com Valor de Tempo:
      count = 1
      STORF(NSTORF+count-1) = TIME
      count = count + 1
!
!
! ------------------------------------------------
! Entrando com os valores da matriz de variáveis
! (REAL(2))

      STORF(NSTORF+count-1) = $Ein
      count = count + 1
      STORF(NSTORF+count-1) = $Iin
      count = count + 1

! ------------------------------------------------
! Foi enviada ao MATLAB as variáveis Ea e Ia de
! acordo com a entrada de dados disponível no
! circuito montado
! ------------------------------------------------
! ------------------------------------------------
! Chamando a interface PSCAD/EMTDC:
! CALL MLAB_INT("MFILEPATH","MFILENAME","Input Format","Output Format")
! ------------------------------------------------
      CALL MLAB_INT("%:Dir\$Path", "$Name", "R(3)", "R")

! ------------------------------------------------
! E possível enviar ainda Variáveis de Resposta do MATLAB
! Para o PSCAD.
! ------------------------------------------------
! Sai da  (REAL(1))
      $Pout = STORF(NSTORF+count-1)
! ------------------------------------------------
! Atualizando os ponteiros STORx
! ------------------------------------------------
END IF
      count = count + 1
      NSTORF = NSTORF + count - 1
! ------------------------------------------------
```

# APÊNDICE C

## Código desenvolvido no M-FILE para Exemplo 1

```
function [Out] = Example1(in)
%
% Essa funcao vai ler a entrada de dados vinda do PSCAD
% a fim de tratar os dados
% Neste Caso, Plotar gráficos de Tensão e Corrente
%
% Entrada:
%   Variável in de dimensao 3
%
% Saída:
% Variável Out de dimensão 1

global Time Ea Ia


Time(length(Time)+1) = in(1);
Ea(length(Ea)+1) = in(2);
Ia(length(Ia)+1) = in(3);
if rem(length(Ia),500) == 0
    figure(1)
    grid on
    hold on
    plot(Time,Ea*1000)

    plot(Time,Ia*10000,'r')
end
Out = Ea(length(Ea))*Ia(length(Ia))
```

# APÊNDICE D

## Código desenvolvido em Fortran para PSCAD para Exemplo 2

```
#STORAGE REAL:3
! ------------------------------------------------
! ------------------------------------------------
! PSCAD / MATLAB INTERFACE
! Modulo: $Name
! Definindo uma variável inteira "count"
#LOCAL INTEGER count

! Verifica se o parametro "Active" esta igual a 1
      IF($Active.GT.0.9) THEN
!! ------------------------------------------------
! Enviando os parametros das variaveis para o MATLAB
! ------------------------------------------------
! Entrando com Valor de Tempo:

            count = 1
            STORF(NSTORF+count-1) = TIME
            count = count + 1

! ------------------------------------------------
! Definido os parametros das variaveis para o MATLAB
! ------------------------------------------------
! Entrando com os valores da matriz de variaveis
! (REAL(2))

            STORF(NSTORF+count-1) = $Ein
            count = count + 1
            STORF(NSTORF+count-1) = $Iin
            count = count + 1

! ------------------------------------------------
! Foi enviada ao MATLAB as variaveis Ea e Ia de
! acordo com a entrada de dados disponivel no
! circuito montado
! ------------------------------------------------
! ------------------------------------------------
! Chamando a interface PSCAD/EMTDC:
! CALL MLAB_INT("MFILEPATH","MFILENAME","Input Format","Output Format")
! ------------------------------------------------
          CALL SIMULINK_INT("%:Dir\$Path", "$Name", "R(3)")

! ------------------------------------------------
! E possivel enviar ainda Variaveis de Saida do MATLAB
! Para o PSCAD.
! ------------------------------------------------
! Sai da  (REAL(1))
         $Pout = STORF(NSTORF+count-1)
! ------------------------------------------------
! Atualizando os ponteiros STORx
! ------------------------------------------------

            count = count + 1
            NSTORF = NSTORF + count + I_CNT- 1
       END IF
```

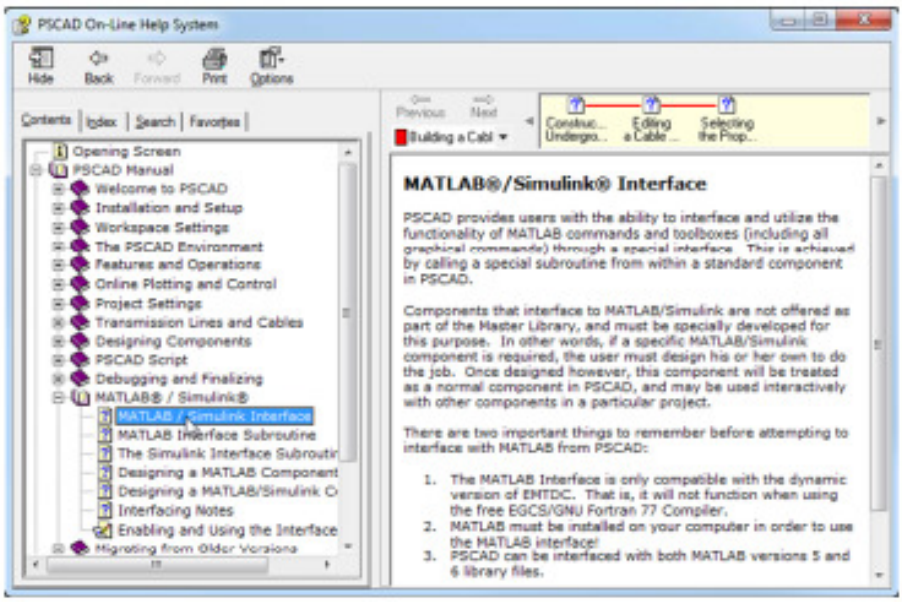
# ANEXO A

# Integração MATLAB – PSCAD

## PSCAD Help

Detailed Matlab procedures can be found in the PSCAD User's Guide or PSCAD On-line help, which can be accessed by pressing F1 in the PSCAD workspace.

Under the Contents tab |PSCAD Manual|"MATLAB® / Simulink®"


PSCAD On-Line Help System
Hide
Back
Forward
Print
Options
Contents
Index
Search
Favorites
Opening Screen
PSCAD Manual
Welcome to PSCAD
Installation and Setup
Workspace Settings
The PSCAD Environment
Features and Operations
Online Plotting and Control
Project Settings
Transmission Lines and Cables
Designing Components
PSCAD Script
Debugging and Finalizing
MATLAB® / Simulink®
MATLAB / Simulink Interface
MATLAB Interface Subroutine
The Simulink Interface Subroutir
Designing a MATLAB Component
Designing a MATLAB/Simulink C
Interfacing Notes
Enabling and Using the Interface
Migrating from Older Versions
Previous
Next
Building a Cabl
Editing a Cable ...
Selecting the Prop...
MATLAB®/Simulink® Interface
PSCAD provides users with the ability to interface and utilize the functionality of MATLAB commands and toolboxes (including all
by calling a special subroutine from within a standard component in PSCAD.
Components that interface to MATLAB/Simulink are not offered as part of the Master Library, and must be specially developed for this purpose. In other words, if a specific MATLAB/Simulink component is required, the user must design his or her own to do the job. Once designed however, this component will be treated as a normal component in PSCAD, and may be used interactively with other components in a particular project.
There are two important things to remember before attempting to interface with MATLAB from PSCAD:
1. The MATLAB Interface is only compatible with the dynamic version of EMTDC. That is, it will not function when using the free EGCS/GNU Fortran 77 Compiler.
2. MATLAB must be installed on your computer in order to use the MATLAB interface!
3. PSCAD can be interfaced with both MATLAB versions 5 and 6 library files.

## Setup

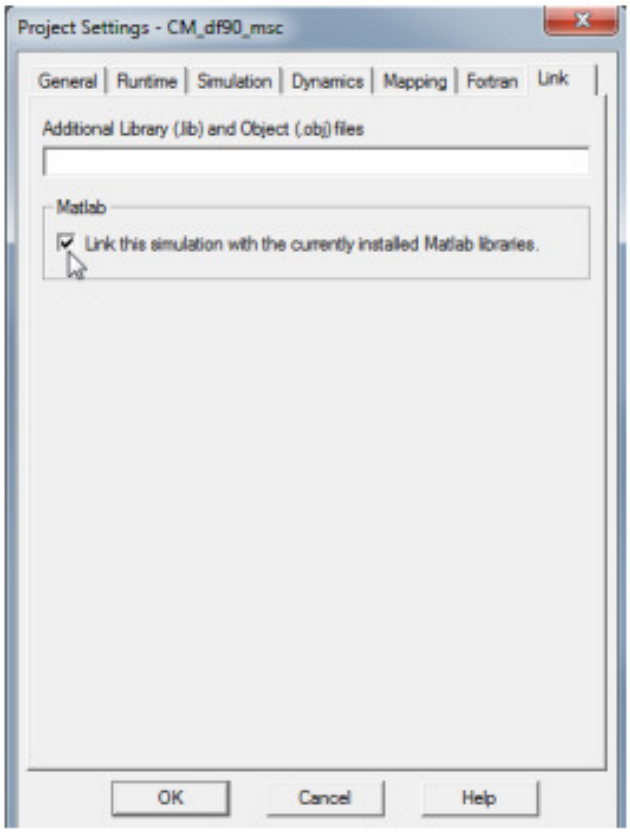
Before you can begin, please verify whether the following requisites are met on your computer:

1. MATLAB has to be installed prior to PSCAD. You may need to reinstall PSCAD to make it recognize the links and dependencies to MATLAB, its libraries and paths.
2. The MATLAB interface feature is not supported by the GNU Fortran compiler, which is provided with your PSCAD package. It is recommended that you install **Compaq Visual Fortran 6.0** or higher or **Intel Fortran Compiler**.
3. Please note that the user has to create a user-defined component to use the MATLAB interface feature; i.e. you will not find a "MATLAB Interface" component in the Master Library in PSCAD.
4. Creating new components requires an Educational or a Professional license.
5. Before starting to run a MATLAB interface simulation, you should direct PSCAD to MATLAB's shared library folder. The shared libraries of MATLAB R200X are located at:

   %MatlabRoot%\extern\lib\win32\microsoft

In order to specify this path for PSCAD, select Version 5 under Workspace Settings | MATLAB | Installed Version. Then enter the **complete** path of the shared libraries. (e.g. C:\Program files\Matlab\R2008a\extern\lib\win32\microsoft) under 'Library Path'.

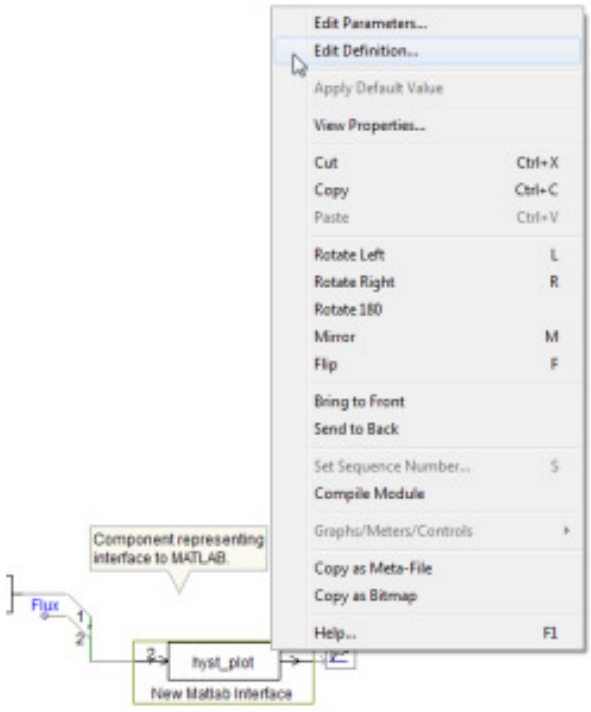
Also, if you are creating a new model, which uses the MATLAB interface function, you have to link your model to the installed MATLAB libraries by activating the corresponding checkbox on Project Settings | Link:



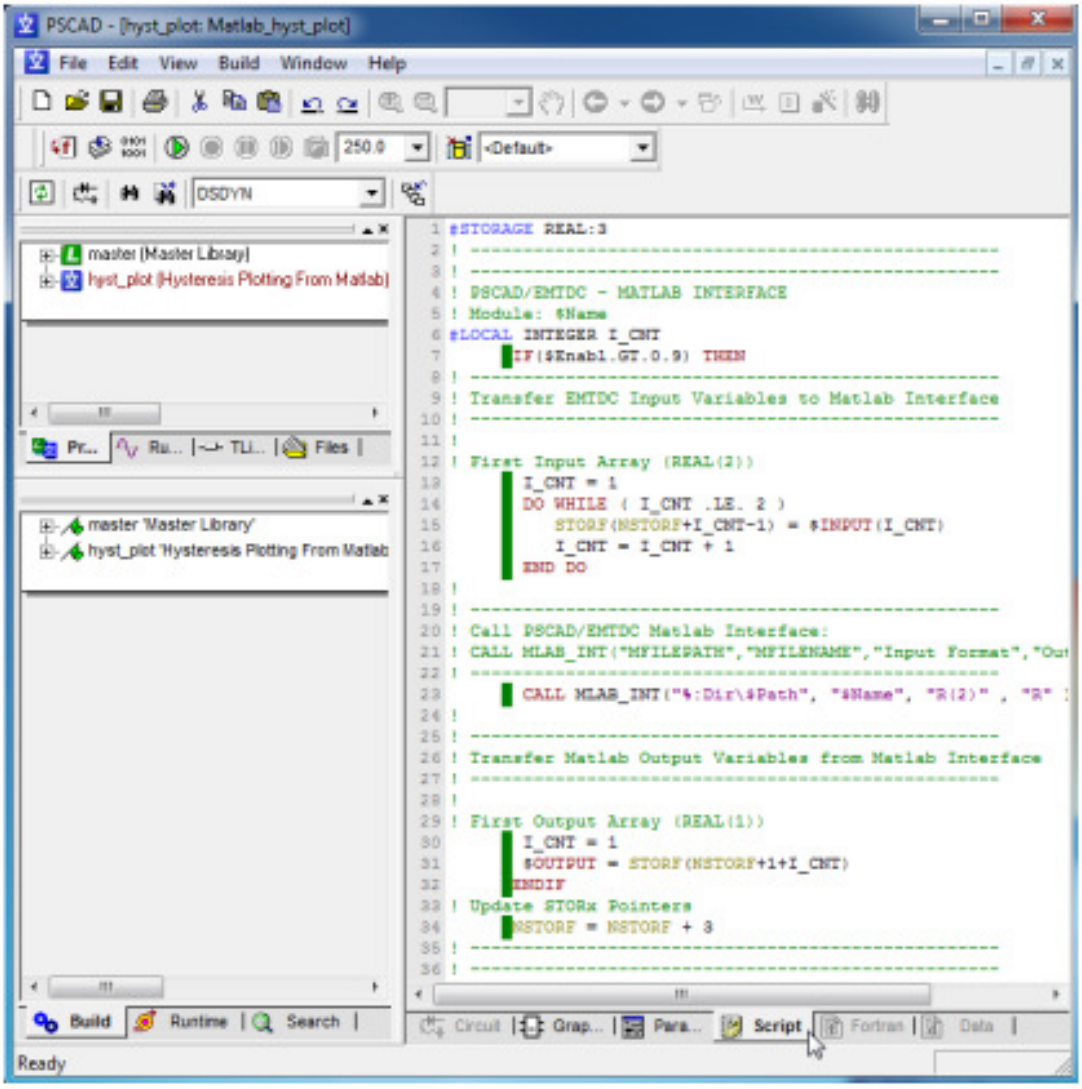

This has already been activated in the examples in %PSCADfolder%\examples\matlab.

A few examples are provided in "%*PSCAD FOLDER*%\examples\matlab". In each of those cases, locate a block specified as "New Matlab Interface".

By right clicking on the component and selecting the "Edit Definition ..." option (as shown below), you will be able to explore the way the interface has been designed in each particular example.




You will see the PSCAD Script code of each block on the "Script" pane of the component definition window:


PSCAD - [hyst_plot: Matlab_hyst_plot]
File Edit View Build Window Help
master (Master Library)
hyst_plot (Hysteresis Plotting From Matlab)
Build Runtime Search
Circuit Grap... Para... Script Fortran Data
Ready

The key subroutine for using the MATLAB interface feature is 'MLAB_INT' (see PSCAD/EMTDC User's Guide). Also, proper use of 'STORF' and 'STORI' is required for exchanging variables between EMTDC and the MATLAB workspace.

I personally found the user's guide and the provided examples very helpful when I was trying to create my first *MATLAB Interface* case. So, please first try the instructions and examples. I recommend that you explore the corresponding script code blocks in the given examples. If you still encounter ambiguities, we will be glad to assist you.

# ANEXO B

## Código em Fortran para Pscad para Exemplo do Suporte Técnico

```
#STORAGE REAL:12
#LOCAL INTEGER I_CNT
#LOCAL INTEGER M_CNT
!-----------------
!    PSCAD/EMTDC MATLAB INTERFACE
!    MODUAL : Interface  Matlab com Simulink
!
       I_CNT=1
       M_CNT=0
!

!-----------------
!Caso M-File
       #IF ($OPTSEC == 0 )
       STORF(NSTORF)= TIME
       STORF(NSTORF+1)= DELT
       M_CNT = 2
!
!Caso Simulink
       #ELSE
       STORF(NSTORF)= TIME
       M_CNT = 1
       #ENDIF

!-----------------
!Armazenando valores dos dados de entrada
       DO I_CNT = 1, $#DIM(sig_in)
         STORF( NSTORF + M_CNT + I_CNT - 1 ) = $sig_in( I_CNT )
       END DO
       M_CNT = M_CNT + I_CNT - 1
!
!---------------------
! CHAMANDO A INTERFACE PSCAD/EMTDC MATLAB
!---------------------
       #IF $OPTSEC == 0
       CALL MLAB_INT("%:Dir\$Path","$mfile","R(6)","R($#DIM(sig_out))")
       #ELSE
       CALL SIMULINK_INT("%:Dir\$Path","$simfile","R(5)")
       #ENDIF
!
!---------------------
! Transferindo as Variáveis de Saída do MATLAB
!
!
       I_CNT = 1
       DO WHILE (I_CNT .LE. $#DIM(sig_out))
         $sig_out(I_CNT) = STORF( NSTORF + M_CNT + I_CNT - 1)
         I_CNT = I_CNT+1
       END DO

!
!-----------------
! Atualizando Ponteiros
       NSTORF = NSTORF + M_CNT + I_CNT - 1
!---------------------
!---------------------
```